International Journal of Innovative Research and Advanced Studies (IJIRAS)
Volume 3 Issue 6, May 2016

ISSN: 2394-4404

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.

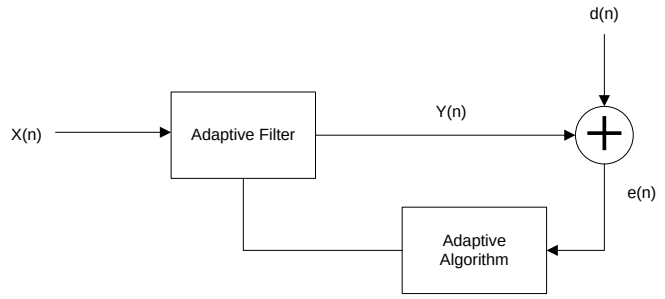

Figure 1: Block Diagram of Proposed Work

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 weighs of the filter tap are updated according to the following formula given by,

$$W(n+1) = W(n) + > \mu . e(n) . x(n) \quad (1)$$

$$e(n) = d(n) - y(n) \quad (2)$$

$$Y(n) = x(n).\ W^{T}(n) \quad (3)$$

Where x (n) denotes the input vector, μ 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,

$$y = \sum_{k=0}^{N-1} W_k . x_k \quad (4)$$

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,

$$W_k = -W_{z0} + \sum_{l=1}^{L-1} W_{kl} . 2^{-1} \quad (5)$$

Where W_k represents the Lth bit of W_k,

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

$$y = \sum_{l=1}^{L-1} 2^{-1} . \left[\sum_{k=0}^{N-1} x_k . W_{kl} \right] + - \sum_{k=0}^{N-1} W_{k0} . X_k \quad (6)$$

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

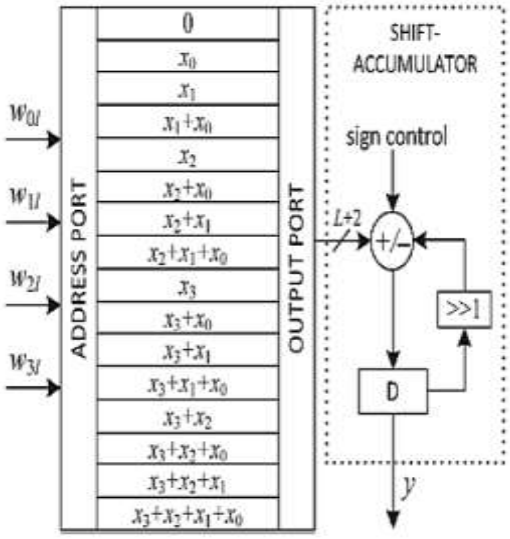

Figure 2: Traditional DA based Implementation

Because the shift accumulation in as in Figure 2 encompasses massive imperative course, it's performed using carry save accumulator, as shown in Figure 3. The bit slices of vector c are fed one after the other within the LSB to the MSB order to the CSA. Nonetheless, the negative (two's complement) of the LUT output is required to be gathered in case of MSB slices. So, the whole bits of LUT output are handed through XOR gates with a sign managed input which is ready to '1' only when the MSB slice appears as address. The XOR gates as a result produce the one's complement of the LUT output corresponding to the MSB slice however do not have an effect on the output for different bit slices. In the end, the sum and the carry words that are bought after L clock cycles are primary to be brought via a final adder which has been excluded from the Figure and the input carry obtained of the final adder is required to be set to '1' to account for the 2's complement operation of the LUT output corresponding to the MSB slice. The content for the k_{th} LUT location is given in the equation below,

$$C_k = \sum_{j=0}^{N-1} x_j . k_j \quad (7)$$

ISSN: 2394-4404

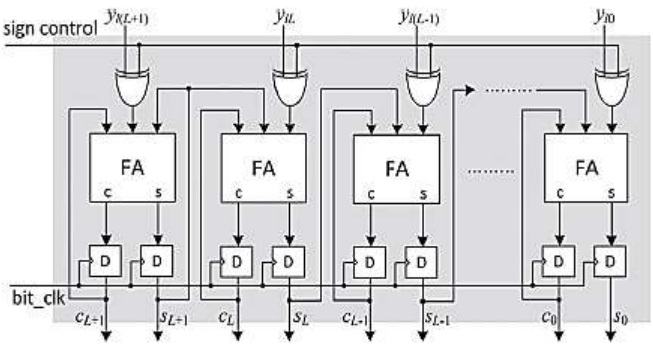

Figure 3: Design for CSA

Where k_j denotes the (j + 1) th bit of N-bit binary illustration of integer k for $0 \le k \le 2^N - 1$. Word that C_k for $0 \le k \le 2^N - 1$ 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 = four. 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.

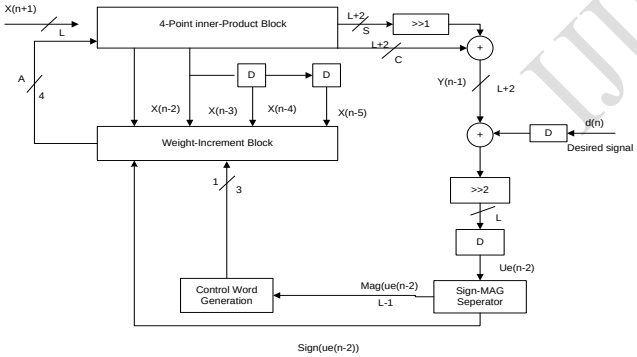

Figure 4: Proposed Structure of Adaptive LMS Filter of Filter Length N = 4

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.

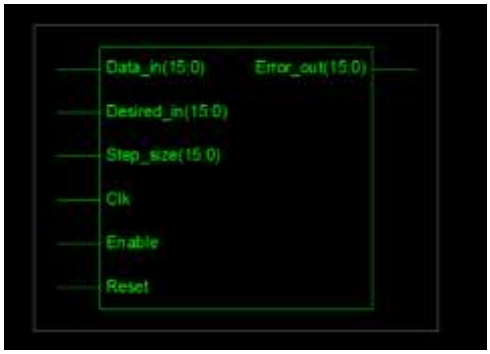

Figure 5: Chip Level Diagram for the Proposed System

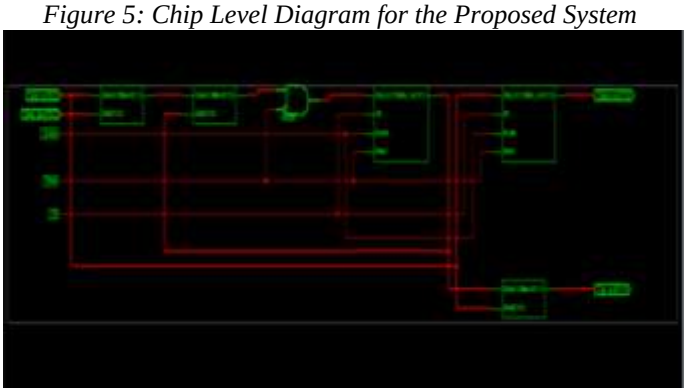

Figure 6: LMS Design using CSA Adder

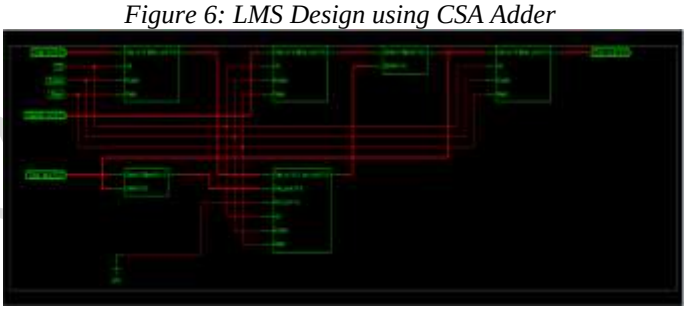

Figure 7: Building Block of CSA Adder

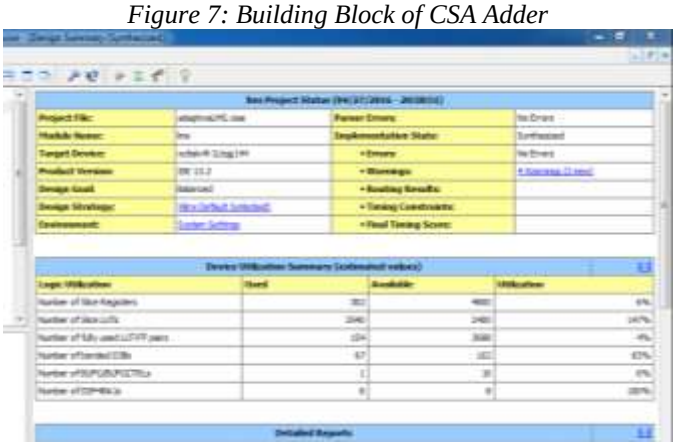

Figure 8: Design Summary

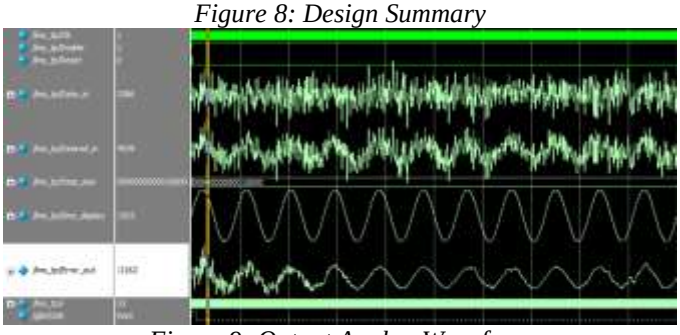

Figure 9: Output Analog Waveform

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

[1] S. Subathradevi and C. Vennila, "Modified Architecture for Distributed Arithmetic with Optimized Delay using Parallel Processing", Indian Journal of Science and Technology, Vol. 8, Issue 24, 2015.
[2] Kairoju Ramachary and L. R. Siva, "Low-Power, High-Throughput, and Low-Area Adaptive FIR Filter Based on Distributed Arithmetic", International Journal of Scientific Engineering and Technology Research, Vol.04, Issue.25, 2015.
[3] Lisha Anna Daniel and Niji Mathews, "Area Efficient Implementation of Adaptive Fir Filter Based On Distributed Arithmetic", International Research Journal of Engineering and Technology (IRJET), Vol. 02, Issue 09, 2015.
[4] Wasim Maroofi,Lalit Jain,"Distributed Arithmetic based Low-Power LMS Adaptive FIR Filter Design", International Journal of Computer Applications, Vol. 132 , No.16, 2015.
[5] Mayur B. Kachare and Prof. D. U. Adokar, "Realization of Distributed Arithmetic (DA) based reconfigurable digital FIR filter", International Journal of Research in Advent Technology, Vol.3, No. 10, 2015.
[6] G. Swathi and M. Revathy, "Design of a Multi-Standard DUC Based FIR Filter Using VLSI Architecture", International Journal of Scientific Engineering and Research (IJSER), 2014.
[7] G. Sathiyavani, J.Amali and S. Indumadhi, "Implementation of Adaptive FIR Filter for the Wavelet Transforms Using Distributed Arithmetic Technique", Vol. 4, Issue 3, pp. 05-08, 2014.
[8] M.Usha and R. Ramadoss, "An Efficient Adaptive Fir Filter Based On Distributed Arithmetic", International Journal of Engineering Science Invention, Vol. 3, Issue 4, PP.15-20, 2014.
[9] C.K.Bagyasri, "Implementation of Efficient FIR Filter Using Distributed Arithmetic", SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE), Issue 8, 2014.
[10] Jyothirmayi Alahari and M.Valarmathi, "Optimized Adaptive Fir Filter Based On Distributed Arithmetic", International Journal of Science, Engineering and Technology Research (IJSETR), Vol. 3, Issue 4, 2014.