Design of Adaptive FIR filter for Biomedical Signal Processing Applications

G.Sahaana
Department of Electronics and Communication Engineering
Veltech Multitech Dr.Rangarajan Dr. Shakunthala Engineering College
Avadi,Chennai,India
sahaanaji1996@veltechmultitech.org

Abstract—**Emerging technologies in DSP systems require high performance in order to provide optimal performance. In addition, Adaptive Filtering plays a key role in the implementation of a variety of digital signal processing algorithms and it can change their characteristics to achieve desired output. VLSI implementations of these designs are efficiently used for filtering and communication. This project proposes design of Finite Impulse Response (FIR) Adaptive Filter which employs distributed arithmetic (DA) circuits for availing improvement in performance. DA circuits explicitly reduces the number of partial product generation. That is existing Design is modified to reduce the delay and hardware requirements. In modified filter, the number of adders and multipliers are reduced which effectively enhances the speed of computation. The VLSI design for 8, 16, 32 and 64 taps are analyzed and compared. These designs are implemented using Altera Quartus II software with family stratix II and device EP2C70F896C6 and the results are reported. The results shows that the Filter design using distributed arithmetic circuits offers good performance when compared with Conventional method.**

Keywords-Finite Impulse Response Filter (FIR Filter, Error Computation Module (ECM), Weight Update Module (WUM), ElectroEncephalogram (EEG) .

I. INTRODUCTION

An adaptive filter is a computational device which derives the between two signals in concurrent and in an sequential manner. It will work efficiently on an unknown environment and it is used to track the input signal of time-varying mannerisms. Adaptive algorithms are studied in fields such as speech analysis, communication system analysis, radar or biomedicine, requires the optimal filter or system coefficients need to be adjusted over time depending on the input processing of signal in many real world applications.

To design an efficient filter number of computations should be reduced so the multiplier-less partial product generator is designed to increase the speed [2-3]. The Conventional Adaptive filter uses more number of additions and multiplications in order to reduce the counts of partial product terms in booth multiplier is implied[4].

Distributed Arithmetic plays a major role in reducing the complexity of the computation so the count of partial products is reduced by using DA with efficient adders [5] and LMS algorithm is used for the update of filter coefficients [6].The efficient type of transform called Hopping Discrete Fourier Transform [11] is designed for software radio applications. Filters are quite useful for video compression based applications [12]. Several adders including carry look ahead adders found applications in filtering [13].

In this proposal, DA is merged with Booth recoding algorithm with approximate model for a efficient adaptive filter design. The latency of the design is reduced and the number of logic elements used is effectively optimized by this method.

The rest of the paper is organized as follows. The Adaptive filter and its algorithm is discussed in detail in Section II. The proposed architecture using DA is discussed in detail in section III. The software results have been discussed in section IV. Finally, conclusions and remarks were provided.

II. FIR ADAPTIVE FILTER ARCHITECTURE

In this section, structure and derivation of FIR adaptive filter is discussed. Adaptive filter is a computational device that attempts to model the relationship between two signals in real time in an iterative manner.

Adaptive filtering technique is used to change the filter coefficients or the weights of the filter in order to decrease the error signal produced during computation.

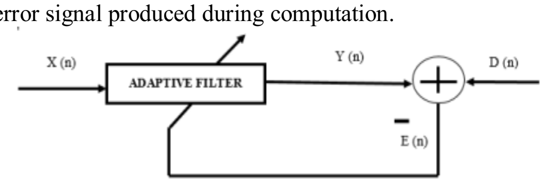


Figure.1 Design of Adaptive Filter [1]

Figure.1 gives the design of adaptive filter where X (n) denotes the input signal, Y (n) denotes the output signal, E(n) denotes the difference or error signal and D(n) denotes the desired signal.

The input signal or sequence, X (n) is given in equation (1).

$$X(n) = [X(n)\ X(n-1) \cdots X(n-L+1)]^T \ldots(1)$$

The Output signal for adaptive filter is predicted by equation (2).Y(n) can also be defined by equation (3).

$$Y\,(n)=\sum_{i=0}^{L-1} w_{i(n)} X\,(n-i) \quad \ldots (2)$$

$$Y\,(n)=w^{T(n)} X\,(n) \quad \ldots (3)$$

where, w(n) – parameter or coefficient vector.

2022 International Conference on Computer, Power and Communications (ICCPC) | 979-8-3503-9784-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCPC55978.2022.10072255

The difference or error signal can be computed using equation (4).

E (n)=D (n)-Y (n)… (4)

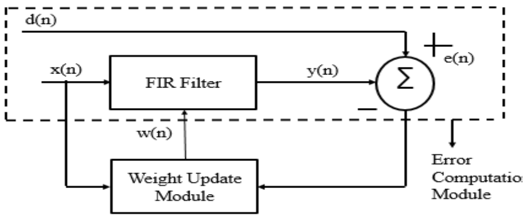


Figure. 2 Structure of Finite Impulse Response Adaptive Filter

Figure 2 shows the structure of Finite Impulse Response Adaptive filter which have two modules namely Error Computation Module (ECM) and Weight Update Module (WUM). ECM produces the error output by comparing the output with the desired signal. WUM updates the filter coefficients to reduce the error produced in ECM and update weight accordingly. Figure 3 and 4 gives the general block diagram of existing ECM and WUM model respectively.

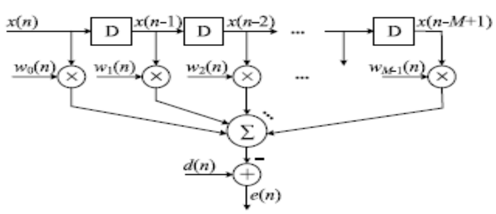


Figure. 3 Existing design of Error Computation Module [1]

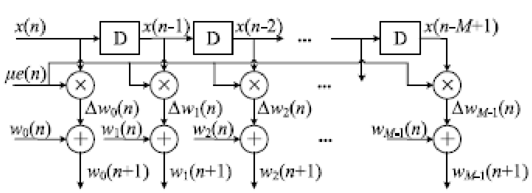


Figure. 4 Existing design of Weight Update Module [1]

A. Least Mean Square ALGORITHM

Least mean square algorithm is an another form of adaptive filter used to reproduce a desired filter by finding the filter coefficients that relate to producing the LMS of the error signal (difference between the desired signal and the output signal).The LMS algorithm for p-th order can be summarized in equation (5). LMS algorithm is an efficient algorithm, which is used to compute the weight to be updated in each iteration. This algorithm can be formulated as equation (5). It is given in detail in [1].

$$w_i(n+1) = w_i(n) + \mu.e(n).x(n-i), i = 0,1,2,\dots,M-1 \dots (5)$$

III. Proposed DA based adaptive filter design

DA is a high - performance computation structure for DSP. It is mainly used for the function of computing sum of products or inner products [6]. DA replaces the multipliers by the Read Only Memory (ROM) Look-up Tables (LUT) and an efficient technique to implement in Field Programmable Gated Arrays (FPGA).Area savings by using DA can be up to 80% in DSP hardware designs. Here, Radix 8 booth encoding with Wallace trees are designed to reduce the partial products and complexity.

A. Error Computation module

ECM in adaptive filter is used to calculate the error signal by comparing the characteristics of desired and present signal. For easy analysis, w_i *(n)* acts as an integer, by shifting it can be easily transformed to fixed-point range. By using the radix-8 Booth encoding or recoding table, as shown in Table 1, four bits of $w_i(n)$ are joined with one overlapping bit. Figure 5 gives the general block diagram of proposed ECM. The weight $w_i(n)$ is computed by the overlapping bit by equation (6).

$$w_i(n) = \sum_{j=0}^{[\frac{m}{3}-1]} \overline{w_i}^{-j}\, 2^{3j} \qquad \dots (6)$$

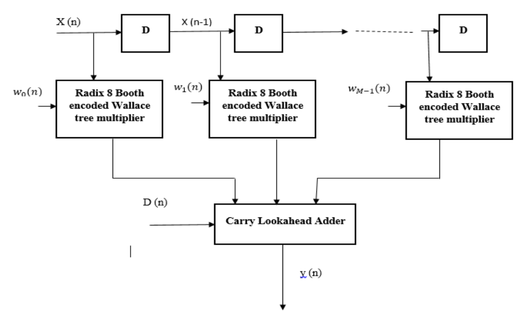
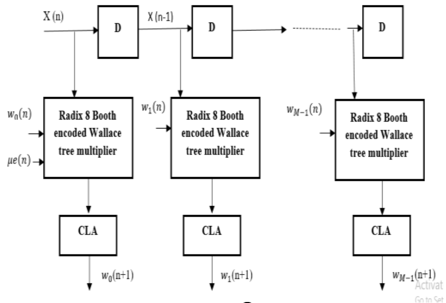


Figure.5 Block Diagram of proposed ECM

B. Weight Update Module

In adaptive filter, weight update is given by following steps , *μ.e(n)*is first computed with a error compensation which is truncated by right shifting. In this case, *e(n)* is represented as an integer for easier analysis, it can be easily transformed to a fixed-point by shifting. If the step size μ for the weight update is$2^{-q}$and q can be a positive integer (for experiment step size is taken as 0.25).the proposed WUM produces less error compared with the existing one it is very useful for real time applications. Figure 6 gives the proposed block diagram of WUM with radix 8 booth encoded Wallace tree multiplier. It can replace the ROM into LUTs so that it can have less complexity.



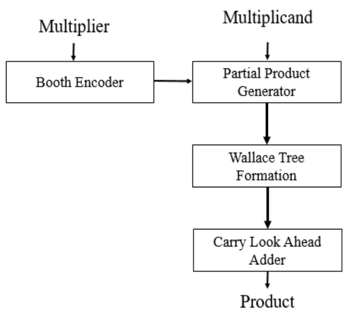


Figure. 6 Block Diagram of proposed WUM
Figure. 7 Block Diagram of Radix 8 Booth encoded Wallace tree multiplier

The internal structure and operation of Wallace multiplier is given in fig 7. Figure 7 clearly gives the operation of the radix 8 booth encoded Wallace tree multiplier. The number of adders used can be given by the Wallace trees produced. For Wallace tree formation Carry Look ahead adder (CLA) is used.

IV. RESULTS AND DISCUSSION

In this section, the comparison between conventional and LMS method of adaptive filter algorithms is given. All outputs are for 8-tap FIR Adaptive Filter. A Conventional adaptive filter shows relatively more complexity than LMS algorithm. LMS algorithm uses recursive method to reduce the complexity. These algorithms for different number of taps are simulated using ALTERA QUARTUS II software and real time application using EEG signal is simulated in MATLAB Simulink software.

The output waveforms are given for reference. Fig.8 is the Waveform for Conventional FIR Adaptive Filter with the input sequence [1 2 3 4 5 6 7] and the output sequence is [8 23 44 70 240 2933 3354 3262 5286 60 38 20 7]. Fig.9 is the Waveform for Error Computation Module using LMS Algorithm with the input sequence [1 2 3 4 5 6 7] and the output sequence [8 23 44 70 100 133 168 140 112 85 60 38 20 7]. LMS algorithm is computed by the step size 0.25. Fig.10 is the waveform for weight update module using LMS algorithm with same input sequence given in ECM. The output is generated for each iteration and it is updated while combining ECM and WUM together. Fig.11 is the waveform for Adaptive filter using LMS algorithm with the input sequence [1 2 3 4 5 6 7] and the output sequence [8 23 44 70 100 133 168 140 112 85 60 38 20 7].

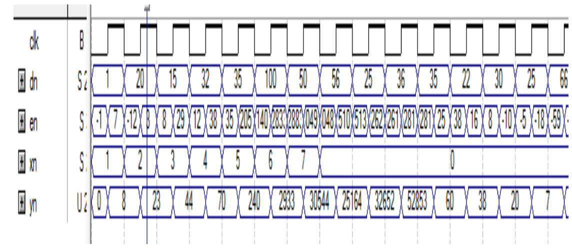


Figure.8Waveform for Conventional FIR Adaptive Filter

Figure.9 Waveform for Error Computation Module using LMS Algorithm

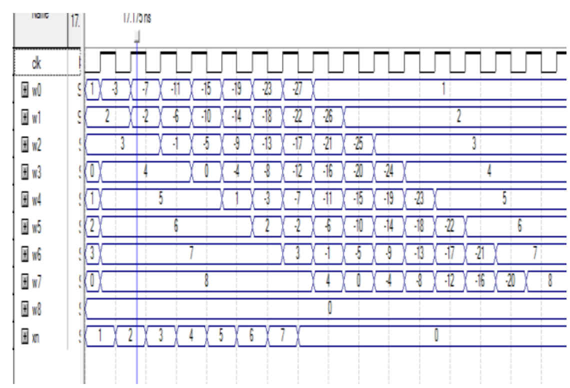


Figure.10 Waveform for Weight Update Module using LMS Algorithm

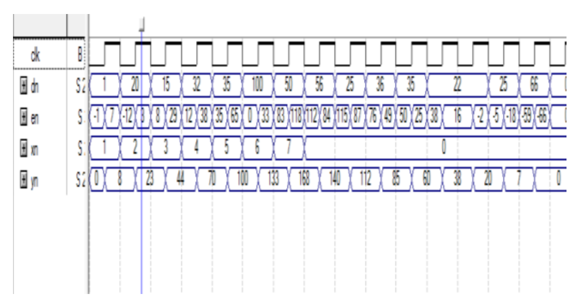


Figure.11 Waveform for FIRAdaptive Filter using LMS Algorithm

Table.2 Comparison table for conventional and LMS method

TYPE	Area (Logic Elements)	Delay (ns)	Power (mw)
Conventional Algorithm	348	22.161	219.87
LMS Algorithm	302	21.520	204.51

Table 2 gives the comparison of conventional and LMS method of FIR Adaptive filter where LMS method shows lower complexity, so it is used for further computation of the proposed method.

Fig.12 is the waveform for proposed ECM using LMS algorithm with the input sequence [1 2 3 4 5 6 7] and the output sequence [8 23 44 70 100 133 168 140 112 85 60 38 20 7]. Fig.13 is the waveform for proposed WUM using LMS algorithm with the input sequence [1 2 3 4 5 6 7] and the output gets updated at each iteration. Fig.14 is the waveform for proposed FIR Adaptive filter using LMS algorithm with the input sequence [1 2 3 4 5 6 7] and the output sequence [8 23 44 70 100 133 168 140 112 85 60 38 20 7].

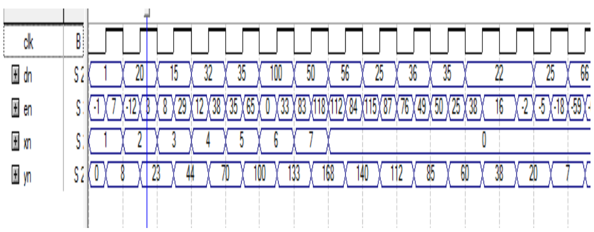


Figure. 12Waveform for Proposed Error computation Module

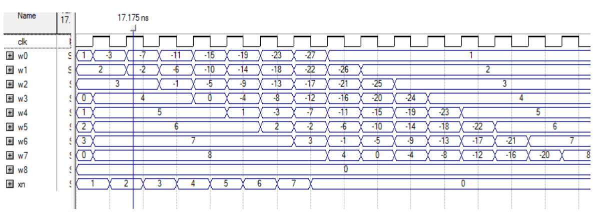


Figure.13 Waveform for Proposed Weight Update Module

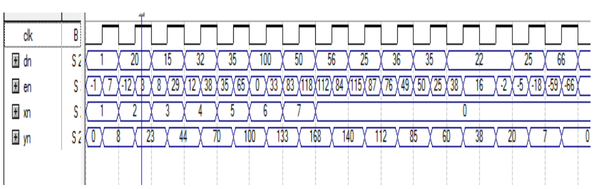


Figure.14 Waveform for Proposed Adaptive Filter

Table.3 Computational complexity for Existing method

Constraints	8-TAP			16-TAP		
Module	**ECM**	**WUM**	**ADAPTIVE**	**ECM**	**WUM**	**ADAPTIVE**
Area	307	144	302	682	300	679
Delay(ns)	8.783	3.754	21.520	7.307	6.475	7.178
Power (mW)	219.41	235.55	219.87	242.53	319 .57	232.64

Table.4Computational complexity for Existing method

Constraints	32-TAP			64-TAP		
Module	**ECM**	**WUM**	**ADAPTIVE**	**ECM**	**WUM**	**ADAPTIVE**
Area	753	150	753	849	253	849
Delay(ns)	11.287	6.909	11.287	8.713	7.143	14.473
Power (mW)	360.56	1643.7	360.97	1631.91	1598.3	1632.0

Table.5Computational complexity for Proposed method

Constraints	8-TAP			16-TAP		
Module	**ECM**	**WUM**	**ADAPTIVE**	**ECM**	**WUM**	**ADAPTIVE**
Area	267	76	97	514	156	549
Delay(ns)	7.193	4.154	8.096	6.542	5.235	3.350
Power (mW)	219.1	225.96	206.64	254.0	301.29	232.34

Table.6Computational complexity for Proposed method

Constraints	32-TAP			64-TAP		
Module	**ECM**	**WUM**	**ADAPTIVE**	**ECM**	**WUM**	**ADAPTIVE**
Area	538	77	538	642	187	642
Delay(ns)	9.861	3.042	7.323	13.751	2.043	9.973
Power (mW)	389.72	1589.8	389.72	1395.0	1433.3	1395.00

Table 3 and 4 tabulates the computational complexity of existing FIR Adaptive Filter for 8, 16,32 and 64 taps. Table 5 and 6 gives the computational complexity of proposed FIR Adaptive Filter for 8, 16, 32, 64 taps. On comparison proposed method shows effective improvement in the area,speed and power. Fig. 15,16 and 17 shows the 64 tap adaptive filter design comparison graph for existing and proposed methods.

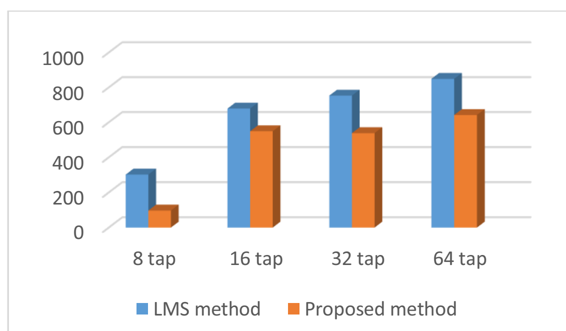


Figure.15 Comparison graph for area

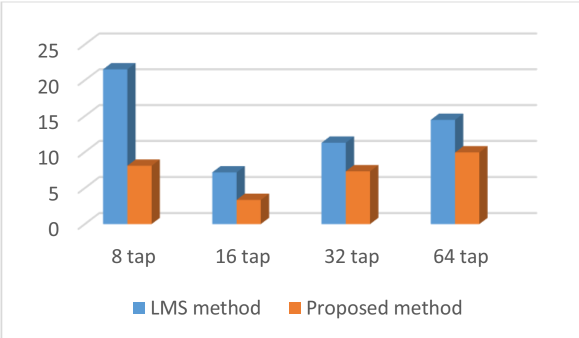


Figure.16 Comparison graph for Delay

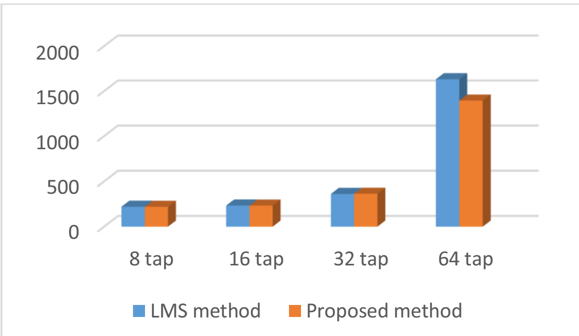


Figure. 17 Comparison table for power

V. CONCLUSION

This paper presents the VLSI Design of a energy efficient and high-performance fixed point FIR adaptive filter design with DA circuits. The results are analyzed using ALTERA QUARTUS II software. The radix-8 Booth recoding algorithm using an booth recoding adder is joined to DA. An Approximate value for partial product is generated and accumulation schemes are proposed for the ECM and WUM modules in the adaptive filter. The hardware risks are significantly reduced due to the use of approximation of DA. The results shows that the proposed method for 64-tap filter offers 24.4% area reduction and 31.1% delay reduction when compared to LMS method.

Future work is to apply the proposed method with the some real time applications like EEG,ECG Analysis.

REFERENCES

[1] Honglan Jiang , "A High-Performance and Energy-Efficient FIR Adaptive Filter Using Approximate Distributed Arithmetic Circuits ", IEEE Trans. On circuits and systems–I: regular papers, vol. 66, no. 1, pp. 313-326, Jan 2019.

[2] Kalyani H , H Umadevi, "Efficient FPGA Implementation of Radix 8 Partial Product Generator for FIR Filter and MAC Applications", International Research Journal of Engineering and Technology (IRJET) ,Volume: 04 Issue: 05, pp. 3095-3098, May 2017.

[3] Xiang et.al., "Analysis on the adaptive filter based on LMS algorithm",Optik-International Journal for Light and Electron Optics (2016), http://dx.doi.org/10.1016/j.ijleo.2016.02.005.

[4] H.Jiang, "Approximate Radix8 Booth Multipliers for Low Power and High Performance Operation", IEEE Trans., vol. 65, no. 8, pp. 2636-2644, Aug 2016.

[5] P. K. Meher and S. Y. Park, "Area Delay Power Efficient Fixed Point LMS Adaptive Filter With Low Adaptation Delay ", IEEE Trans. on very large scale integration (VLSI) Systems, vol. 22, no. 2, pp. 362-371, Feb 2014.

[6] Rui Guo and Linda S. De Brunner ,"Two High Performance Adaptive Filter Implementation Schemes Using Distributed Arithmetic", IEEE Trans. On circuits and systems II:express briefs,vol.58,no.9, pp.600-604, Sep 2011.

[7] F. Abd Rahman and M.F. Othman, "Real Time Eye Blink Artifacts Removal in Electroencephalogram Using Savitzky Golay Referenced Adaptive Filtering", IFMBE Proceedings Vol. 56, pp.68-71, Feb 2016.

[8] Poonam Dhankhar, Suresh Khaleri, "Eye Blink Artifact Removal in EEG Using Adaptive Fir Filter-A Review", International Journal of Emerging Technology and Advanced Engineering , Volume 4, Issue 6, pp. 700-703, June 2014.

[9] A Garcés Correa et al., "Artifact removal from EEG signals using adaptive filters in cascade", J. Phys.: Conf. Ser. 90 012081 , Aug 2007.

[10] Shruthi. R. Patel, "Comparative Study of LMS & RLS Algorithms for Adaptive Filter Design with FPGA", Progress In Science in Engineering Research Journal (PISER), vol. 2, pp. 185-192, Jan 2014.

[11] G.Sahaana, K.Mariammal, "VLSI Implementation of Transforms", Second International Conference on Signals Systems and Communications,-ICSSC, pp-236-239,ISBN -978-81-937750-2-8,Jan 2019.

[12] Banusha Chandran, K.Mariammal, "VLSI Implementation of Frame Compression using efficient Coders", Second International Conference on Signals Systems and Communications,-ICSSC, pp-231-235,ISBN -978-81-937750-2-8,Jan 2019.

[13] T.R Thendral, K.Mariammal, "Design of Configurable Adder and its Application to Filtering", Second International Conference on Signals Systems and Communications,-ICSSC, pp-225-230,ISBN -978-81-937750-2-8,Jan 2019