ANALYSIS OF HIGH ENERGY-EFFICIENTFIR BY USING ADAPTIVE FILTER

[1]B SHINY SUCHARITHA,[2] R ALEKYA ,[3]ALETI SOUMYA

[1,2,3]Assistant professor, ECE Department, St.Martin’s Engineering College,Sec

ABSTRACT

Filters are the most necessary elements of the DSP application systems. Multiply Accumulate Unit (MAC) is the major block in the FIR filter, because of its operation. The complexities of the MAC unit can be reduced by several reduction techniques such as Constant Multiplications (CM) and Distributed Arithmetic (DA) methods. Distributed Arithmetic method replaces the MAC operation by Pre-computed results stored in Look Up Tables. Many DA based techniques have implemented in FIR filter for the reduction of area. Adaptive filters based on least mean square (LMS) algorithm constitute a standard in many DSP applications. The LMS algorithm, being an approximation of the wiener filter, is inherently imprecise, and constitutes a fertile ground to employ approximate hardware techniques with the additional challenge related to the presence of a feedback path for coefficients update. The application ofsystem identification using a 48-tap bandpass filter and a 103-tap high-pass filter shows that the approximate design achievesa similar accuracy as its accurate counterpart. Compared withthe state-of-the-art adaptive filter using bit-level pruning in theadder tree (referred to as the delayed least mean square (DLMS)design), it has a lower steady-state mean squared error and asmaller normalized misalignment. Synthesis results show that theproposed design attains on average a 55% reduction in energyper operation (EPO) and a 3.2× throughput per area comparedwith an accurate design. Moreover, the proposed design achieves45%–61% lower EPO compared with the DLMS design.

1. INTRODUCTION

THE human beings’ superior ability to accurately controlcomplex movements, due to the cerebellum, has engaged considerable attention. Many computational models have beenproposed to explain and to mimic the cerebellar function forsignal processing and motor control applications, includingthe perceptron-based model [1], [2], the continuous spatiotemporalmodel [3], the higher-order lead-lag compensatormodel [4] and the adaptive filter-based model [5]. Amongthem, the most widely used cerebellar model is based on theadaptive filter due to its relatively low complexity and highstructural resemblance to the cerebellum. However, little has been done on implementing the cerebellar model in hardwaredue to its high complexity.

Finite Impulse response filter is a digital filter, whose impulse response is of finite duration [1]. FIR filters does not require any feedback, so the filter is known as non-recursive filters. The output function of the FIR filter is given by

$$Y(n) = X(n) * H(n)$$

For N order filter the each response of output sample is the summation of current values of the input samples.[1]

$$Y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) = \sum_{k=0}^{M-1} x(k)h(n-k)$$

Here, x(n) is the output sequence, h(n) represents the coefficients samples of filter and Y(n) is the output response [1].FIR filters are the essential structural blocks in the signal processing for removing noise from the original sound and it has various advantages like stability, linear phase response, and regular structure[2]. Optimization of area, power, and delay plays a major role in FIR filters, since it is used in VLSI signal processing applications and communication [2]. Multipliers are the major blocks in the FIR filter because major operations and delay took place in this particular block. The execution speed of the FIR filter is decided by the employed multipliers [3].

Many papers on the focused on the reduction of complexity of the LMS algorithm, which is mainly due to the multipliers in the FIR filter and in the coefficient update circuitry (fig.1). In [6] a CORDIC version of the LMS algorithm is proposed, allowing to replace about fifty percent of multipliers with pipelined CORDIC units. The usage of distributed arithmetic (bit-serial operations and LUTs) is examine in [7]-[9] to reduce area occupation and power dissipation. A critical part analysis of the LMS algorithm is presented in [10]. Here the

authors observed that, for the most practical cases, no-pipelining of the LMS algorithm is required, while, when high sampling rates are required (e.g. radar applications), delay-LMS (DLMS) can be adopted. In the multiplication operations are realized has shit-and-add operations by adopting the SPT format, in the context of the sign-LMS family.

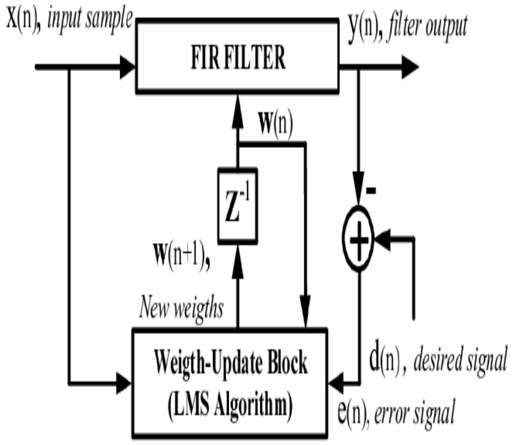


Fig 1: Adaptive LMS filter.

The LMS is a recursive algorithm aimed to minimize the mean-square-error (MSE) between the FIR filter output and a desired signal. The minimization requires MSE gradient computation, which, in the LMS algorithm is computed in an approximated way, causing the so called gradient noise [5]. To the best of authors knowledge, this is the first time that approximate circuits, like are investigated in the context of adaptive LMS filtering.

The applications in system identification and the saccadicsystem show that the proposed approximate FIR adaptive filters incur a very small loss in accuracy compared withthe accurate implementation. Synthesis results indicate thatthe proposed design achieves nearly 55% reduction in energyper operation (EPO) and a 3.2× throughput per area (TPA).Compared with the delayed least mean square (DLMS)-baseddesign of [7], the proposed design requires up to 60% lowerEPO with a higher accuracy (i.e., lower mean squared errorand misalignment).

2. LITERATURE REVIEW

Guo and DeBrunner (2011a), Guo and DeBrunner (2011b) have modified the DA-based computation of LMS filter where a single LUT is used instead of two LUTs as required by the structure proposed by Allred et al. (2005) for both filtering and weight updation. The structures of [Allred et al. (2005); Guo and DeBrunner (2011a); Guo and DeBrunner (2011b)] uses RAM-based LUTs and involve several clock cycles for updating the LUT contents which increase the iteration period. Although the DA-based structures of LMS filter proposed by [Allred et al. (2005); Guo and DeBrunner (2011a); Guo and DeBrunner (2011b)] involve less area than the corresponding multiplier-based structures of LMS adaptive filter.

The iteration period of DA-based structures still considered to be higher. Few attempts also have been made to develop DA structures for DLMS algorithm. Park and Meher (2013) have proposed a pipelined DA structure for DLMS adaptive filter, where a single LUT is used for both filtering and weightupdating operations similar to the structure proposed by Guo and DeBrunner (2011a). But the structure of [Park and Meher (2013)] uses a register-based LUT instead of RAM-based LUT. The register-based LUT requires one clock cycle to update its contents. Also, carry save accumulation method is used in the structure proposed by Park and Meher (2013) to reduce the iteration period of the DA-based structure of LMS filter.

DA-based structures of LMS adaptive filter proposed in [Guo and DeBrunner (2011a); Guo and DeBrunner (2011b); Park and Meher (2013)] are efficient and suitable for real-time applications. But these structures processes one sample in every iterations, and they offered fixed sampling rate for fixed clock frequency where the clock frequency is constrained by the iteration period of the design and the technology node used for hardware implementation of pipeline structures. The block LMS (BLMS) adaptive filter is a useful derivative of the LMS adaptive filter. It offers L fold higher sampling rate than the LMS-based adaptive filter and involve less than L times more computation than those required by LMS adaptive filter, when convolution and correlation operations are performed using fast Fourier transform (FFT)/ inverse fast Fourier transform (IFFT) [Clark et al. (1981)].

BLMS adaptive filter processes a block of input and computes a block of error which is used for updating the weight-vector for the input block. The error performance of the BLMS algorithm is similar to that of LMS algorithm, but the BLMS adaptive filter offers L fold higher throughput compared with the LMS adaptive filter at the cost of nearly L times more computations, where L is the block length. BLMS algorithm is more popular for software implementation of adaptive filter for various practical applications [Patra et al. (1999)] as FFT/IFFT are implemented more conveniently in software. Shen and Spanias (1996) proposed frequency-domain block filtered-XLMS (BFXLMS) algorithm for the block filtering using FFT/IFFT to reduce the computational complexity. In FFT/IFFT algorithm, the data feeding pattern changes after every butterfly stages which is referred as irregular data flow. The FFT/IFFT algorithm in volves butterfly structures with irregular data-flow which are not favorable for very large-scale integration (VLSI) implementation. In the recent study, it has been observed that the time-domain BLMS algorithm offers some redundant computations which could be avoided using DA scheme to develop efficient hardware structures.

3. DA TECHNIQUES IN ADAPTIVE FILTERS

Adaptive filters are widely used in the applications of Echo cancellation, communication, networking and signal processing [14].The main difficulty of implementing DA in adaptive filter design is the updating of coefficients at every clock cycles. D.J. Allred et al implemented a new DA based Least Mean Square technique in Fir filter. This technique has reduced power dissipation by decreasing the clock speed with increase in area by implementing the auxiliary LUT. This auxiliary LUT is mainly used to updating the contents of LUT used in adaptive FIR filters. The proposed architecture is composed of three modules namely DA filter module, auxiliary LUT and the controller module.

Rui Guo et al have implemented two Distributed Arithmetic based techniques in which the Adaptive FIR filter is employed for the reduction of the hardware complexity. The first method uses the commutative property of convolution process for filtering operation and for address of the LUTS.

The second method uses object binary coded method for the decrease of LUT entries [16]. This two methods has eradicated the need of auxiliary LUT for updating the contents, hence the area requirement for implementation of this algorithm is reduced than the previous methods. The process of weight adaptation is a complex process leads to the difficult mathematical procedures, because no partial products of the filter coefficients are stored. The pipeline Architecture method has replaced the adder/ shift accumulation block of DA algorithm by Carry save accumulation unit. Two different Clock periods are used in this design. One clock period is used for carry save accumulation block and another clock period is utilized by remaining all other devices in the circuit. The architecture proposed by Park.et al has reduced both power and area requirement. The Look up table sharing technique for the computation of outputs and weight increment terms has reduced the usage of adders in the algorithm.

The area and power requirement of various algorithms is also analyzed. The algorithm proposed by Park et al has lowest power requirement, whereas the algorithm proposed by Mohanthy et al has lowest area requirement.

Harpreet Singh et al have implemented the DA technique in Adaptive filter. Memory elements are not used for updation of filter coefficients. The one scaling accumulator is used for storing and shifting the sum of all DA base units.

SuganthiVenkatachalam et al have proposed the three approximate sum of products using Distributed Arithmetic technique. The first model affords a notable power reduction and other two models provide better accuracy with increased area and power than the first model [20]. They implemented this DA based ASOP in Image signal processing applications, where the accuracy of the output plays a major role.

4. PROPOSED ADAPTIVE FILTER ARCHITECTURE

For an *M*-tap direct-form FIR adaptive filter (i.e., an *m*-bitfixed-point implementation), the critical path delay is the sumof delays in the error computation $(tM+_log2(M +1)_\times tA)$and weight update processes $(tM+ tA)$, where $tM$and Ta are the critical path delays of an $m \times m$ multiplier and an*m*-bit adder, respectively. Therefore, the sample rate of theinput signal is limited due to this long latency. An importantfeature of the proposed adaptive filter using DA is thereduction of the latency to achieve a high throughput withsignificantly low area and power consumption. In the adaptive learning process for the weight update,errors in the adaptive filter circuit can be inherently compensatedor corrected. Therefore, power and area efficientapproximate arithmetic circuits are considered for a fixedpointimplementation. Truncation is an efficient method tosave power and area for approximate arithmetic circuits ata limited loss of accuracy so it has been extensivelyused in the design of fixed-width multipliers. Mostexisting designs are based on the truncation of the partialproducts to save circuitry for partial product accumulation. All bits of the input operands are required forthese multipliers and therefore, memory is not reduced forstorage requirements. However, memory consumes a significantamount of power and accounts for a large area inan application involving a large data set. Moreover, efficientdata transfers are very important for achieving a highthroughput.

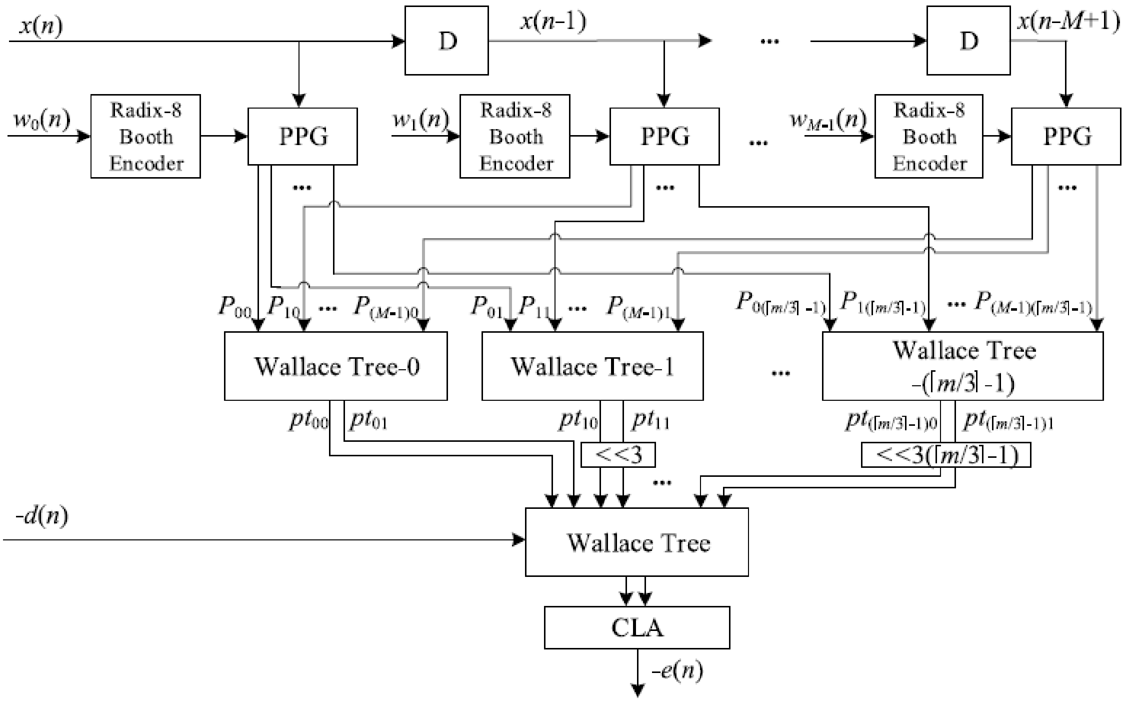


Fig. 2. Proposed error computation scheme using distributed arithmetic. PPG: the partial product generator; CLA: the *m*-bit carry lookahead adder.

As per the results, compared to the partial producttruncation, truncating the input operands achieves more significantreduction in hardware overhead for adder and multiplierdesigns. Thus, truncation on the input operands is applied toachieve savings in the partial product generation.Fig. 2 shows the proposed error computation moduleusing DA. In this design, no LUT is used due to the largesize incurred in a high-order filter. Thus, the partial productvectors *PPi j* are generated online and accumulated.

4.1 Approximate lms adaptive filters:

The paper investigates the error-performance trade-off of approximate adaptive LMS filters, employing the multipliers proposed in a system identification application. The analysis reveals that the choice of the approximated multiplier topology must be carefully examined, otherwise, due to the presence of the feedback path, the stability and convergence performance of the algorithm can be compromised. To this purpose the propose a variation of the multiplier, to optimize the error-performance trade-off. The proposed circuits are implemented, and showing that adaptive LMS filters based on the proposed multiplier allow reduction of power dissipation with tolerable convergence error degradation.

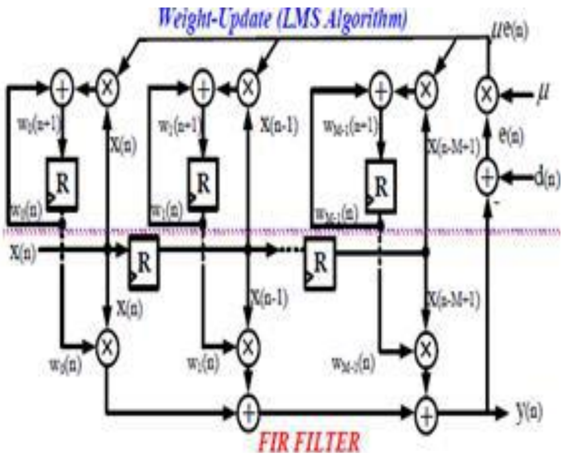
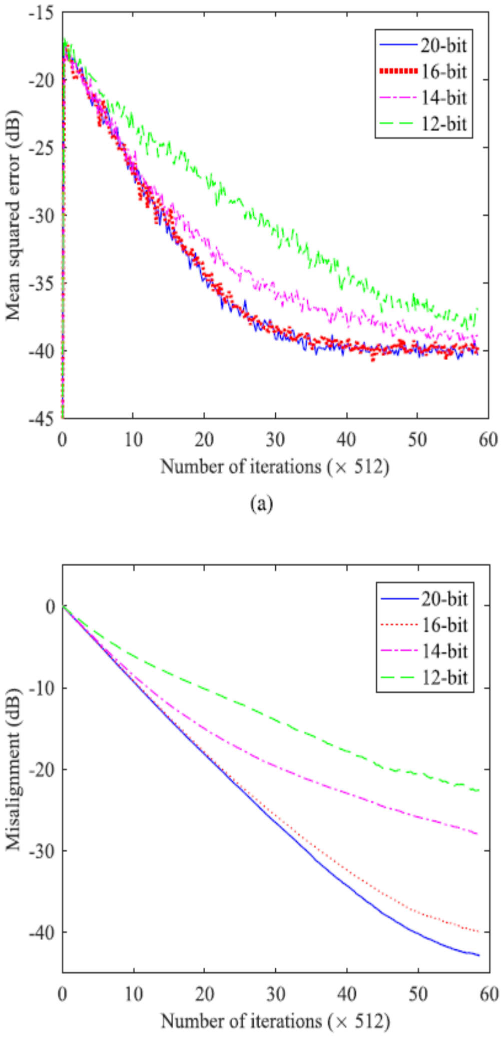


Fig.3. Adaptive LMS filter-hardware overview

4.2Approximate Multiplier:

Multipliers are most important parts in signal processing applications or other computationally drastic applications. Therefore, multiplier designs are mainly focused on high-speed, low area and low power. These parameters are achieved by approximate multipliers. Generally, approximate computing has a significant attention as a rising strategy to decrease power consumption of error tolerant applications like image processing. Approximate computing has been advocated as a new approach to saving area, as well as increasing performance at a limited loss in accuracy. The partial product matrix (PPM) of approximate multipliers consist of approximate 2x2 multiplier block, which takes four partial products, belonging to three adjacent columns of the PPM, and computes an approximated sum, producing three output bits (in place of four). The most significant and the least significant partial products of each block are provided unchanged as output, while the remaining output bit is obtained by OR-ing the two middle partial products (in this way, an OR gate is used in place of a half-adder). The 2x2 block is replicated along the whole PPM to implement the approximate multiplier.Signed multipliers are often needed in practical DSP applications (such as LMS filtering), therefore in this section we review and the approximate unsigned multipliers to the signed case.

5. EXPECTED RESULTS

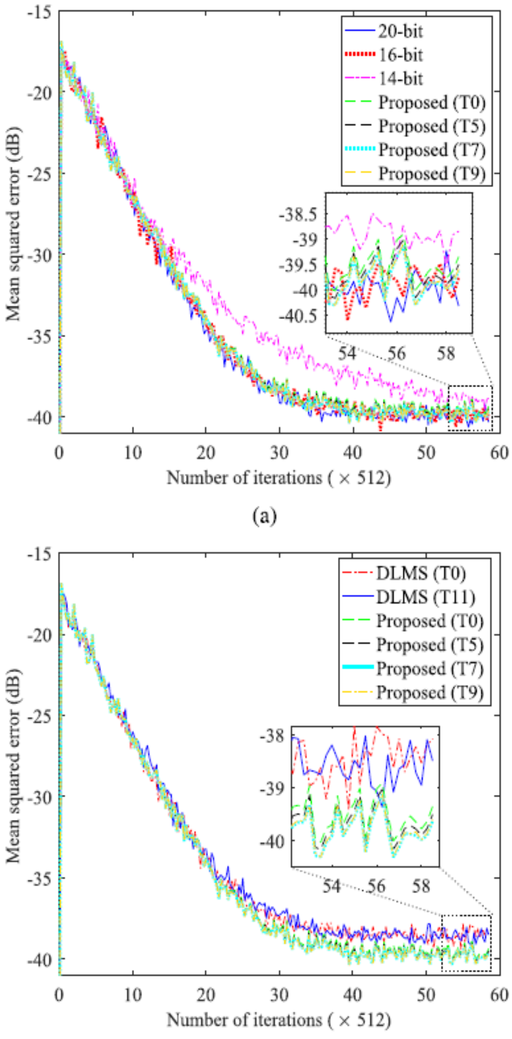


(a)

(b)

Fig. 4. Learning curves of accurate FIR adaptive filters at different

resolutions in (a) the mean squared error and (b) the normalized misalignment.



(a)

(b)

Fig. 5. Comparison of learning curves in the mean squared error

between the proposed 64-tap adaptive filters and (a) accurate implementations

and (b) DLMS-based designs.

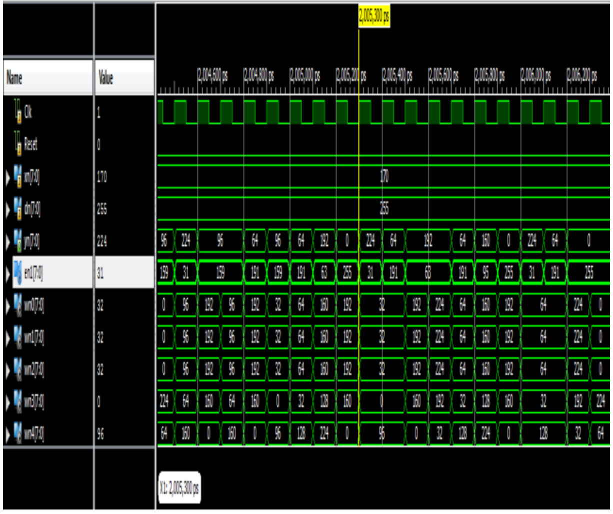


Fig.6. stimulation output

CONCLUSION:

This paper proposes a high-performance and energyefficientfixed-point FIR adaptive filter design. It utilizes an integrated circuit of approximate distributed arithmetic (DA),so it achieves significant improvements in delay, area andpower dissipation. The radix-8 Booth algorithm using anapproximate recoding adder is applied to the DA. Moreover,approximate partial product generation and accumulationschemes are proposed for the error computation and weightupdate modules in the adaptive filter. The critical path andhardware complexity are significantly reduced due to the useof approximate and distributed arithmetic. Adaptive LMS filters using approximate multipliers in the FIR filter section have been investigated for the first time in this paper. The analysis reveals that, due to the feedback loop for the coefficient update, non-aggressive approximate multipliers must be employed. A novel approximate multiplier is proposed, employing optimized partial product approximate compression, least significant columns truncation and mean error compensation.

REFERENCES

1. M.S. Prakash, R.A.Shaik"*High Performance Architecture for LMS Based Adaptive Filter Using Distributed Arithmetic"*, IPCSIT vol. 24 IACSIT Press-2012, Singapore pp18-22

2. S.F. Hsiao, JH ZhangJia, M-C Chen *" Low cost FIR filter designs based on faithfully Rounded truncated multiple constant multiplications*", IEEE Trans. Circuits Syst.-II-2013 Expression Briefs, 60,Page no: 287–291

3. F Nekoei, Y.S Kavian. *"Some schemes of realization digital FIR filters on FPGA for communication applications"*. IEEE Crimean Conference. on Microwave and Telecommunication Technology. September- 2010, Page no. 616–619

4. R.Hartley, "*Subexpression sharing in filters using canonic signed digit multipliers"*, IEEE Transcations-1996.

5. Peled A, B. Liu, "*A new hardware realization of digital filters*", IEEE Transcations on. Acoustic. Speech Signal Processing, volume. ASSP-22 Page no :456-462, 1974.

6. White S. A., *"Applications of distributed arithmetic to digital signal processing: A tutorial review,*" IEEE Transactions -ASSP Mag., volume. 6, page no: 4-19, Jul. 1989.

7. G. N. Jyothi and Sri Devi Sriadibhatla "*Distributed Arithmetic Architectures for FIR Filters-A Comparative review*" IEEE Wi -SPNET -2017 conference- page no: 2684- 2690

8. Ghamkhari S. F.,Ghaznavi-Ghoushchi M. B., "*Low-power low-area architecture design for distributed arithmetic (DA) unit*" 20th Iranian Conference on. IEEE, May, 2012 page no : 15–17

9. C.F. N. Cowan, S.G. Smith, and J.H. Elliott, "*A Digital Adaptive Filter Using a Memory Accumulator Architecture: Theory and Realization*" IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING VOL. ASSP-31, NO. 3, JUNE 1983.Pp 541-549

10. B.Hong, Haibin Y, Xi.Wang, and Ying Xi, "*Implementation of FIR filter on FPGA using DAOBC algorithm*", IEEE -2010.