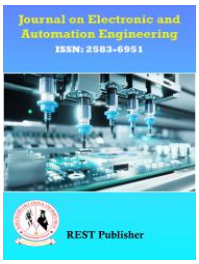
Journal on Electronic and Automation Engineering
Vol: 4(2), June 2025
REST Publisher; ISSN: 2583-6951 (Online)
Website: https://restpublisher.com/journals/jeae/
DOI: https://doi.org/10.46632/jeae/4/2/25



An Area and Power-Efficient Adaptive FIR Filter Architecture for Decision Feedback Equalizers

***Shaik Hussian Ahammed, Karunakar Aliginti**
Sri Sai institute of Science and Technology, Rayachoty, Andhra Pradesh, India.
*Corresponding author Email: hussainahammed610@gmail.com

Abstract: *In this project, a memory less distributed arithmetic based adaptive FIR filter has been proposed for low power and area efficient design. The area and power of the design is decreased, because of using memory less distributed arithmetic architecture in the adaptive FIR filter for computing the inner product. The LUT in the conventional Distributed Arithmetic (DA) based filter is replaced by 2:1 multiplexers in the proposed design to reduce the area. By using enhanced 4:2 compressor adder instead of normal adder, the area of the filter further reduced. The proposed design requires more than half area that required for the existing LUT based inner product block. The proposed design is implemented in synopsis 90nm CMOS technology. The synthesis result shows that the area of the proposed design is reduced by 52.79% when compare with existing architecture. Also, the proposed Adaptive filter causes 69.25% less power consumption for filter tap N=16, 32 and 64. Proposed design provides 36.50% less Area Delay Product (ADP) and 30% less in Power Delay Product (PDP) when compare with the pipelined DA based adaptive FIR filter.*

Key words: *Adaptive Filter, Distributed Arithmetic, weight-increment, FIR, Look-up table, multiplexer, memory-less inner product.*

1. INTRODUCTION

An adaptive filter plays a paramount role in the areas of signal processing application like noise cancellation [1], echo cancellation [2], channel equalizers [3] etc. where only minimum prior knowledge of signal characteristics is available. The basic block diagram of adaptive filter is shown in fig.1. The central part of the adaptive filter is the linear combiner which is used to form the filter output *y* and many adaptive algorithms are present to update the filter coefficient to get optimum values. Windrow-Hoff least mean square (LMS), Normalized LMS, Recursive LS are some of the adaptive algorithms to update the filter coefficient. Windrow-Hoff LMS adaptive filter is more preferable due to less complexity and easy to compute for coefficient update. Adaptive filters can be designed in FIR and IIR. Adaptive FIR filters are more stable and the filter coefficients are easily updated rather than the adaptive IIR filter.

Generally, an adaptive FIR filter contains multipliers and adders to compute the output. The multipliers will occupy more area and consume more power to get rid of this; we go for multiplier less architecture. Many multipliers less architectures are enhanced to perform multiplication operation to consume less area and power consumption. Conversion based methods and memory based methods are the two methods which uses multiplier less design. In conversion based method, filter coefficients are converted into other numerical representation, whose hardware implementation and arithmetic manipulation is more energetic than the binary representation. Canonical signed digit (CSD) and Dempster-Mcleod method are the two conversions based methods in which the filter coefficients are written as the combination of the power of two. Memory based method involves Look Up tables (LUT)/ROM to store the precomputed inner products involved in filtering operation. Constant coefficient and Distributed arithmetic (DA) are significant among the memory based methods.

Distributed Arithmetic (DA) is chosen for implementing adaptive FIR filter, because of its less computing time and area efficient. It is mainly used for calculating sum of products and it occupies less area in DSP application, when filter order increases. DA was first introduced by Croisier [4] later Peled developed the DA architecture [5]. Zohar [6] has done some document work on DA. White [7] [8] proposed an article on vector dot product using

DA and applied the concept on AGM digital autopilot. Allred et al [9] proposed DA based adaptive FIR filter. In the paper [9] the authors used two separate LUT architectures. One LUT is used for calculating filter output and other LUT is used for weight updating of the filter. Brunner and Guo designed DA based adaptive filter by using single LUT for weight update and filtering, but the design is suitable for lower order filters [10]. Park [11] proposed pipelined DA based adaptive FIR filter with low adaptation delay. Mohanty [12] proposed block LMS adaptive filter using DA.

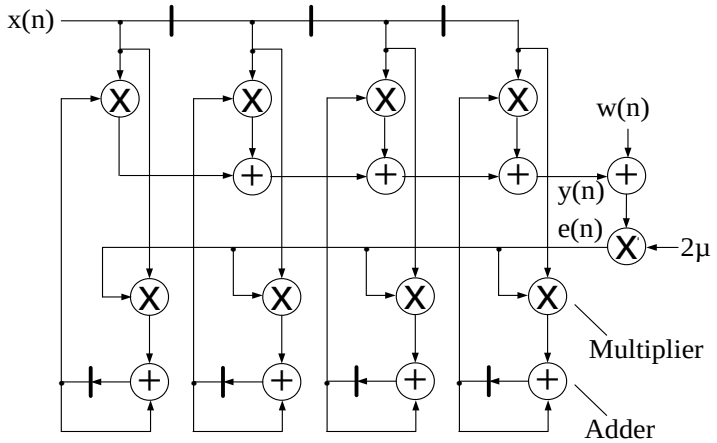


FIGURE 1. Basic Adaptive FIR Filter

The contributions of the paper brief are as follows:

1) Due to usage of multiplexer, instead of LUTs, the area occupied by the proposed adaptive filter architecture is reduced.

2) In proposed design, the input signal and filter coefficients are updated in parallel.

3) Accumulation which is done by conventional adder of DA is replaced by enhanced 4:2 compressor adder to reduce area and complexity of the design.

The organization of this paper is as follows. Section II gives the algorithm of adaptive FIR filter using DA architecture. Section III describes proposed memory-less DA. Section IV describes the four tap adaptive FIR filter architectures based on Memory-less DA (MLDA). Section V describes the 16-tap adaptive FIR filter. Synthesis results of the proposed design are explained in section V. Finally, section VI concludes the paper.

2. OVER VIEW OF LMS ADAPTIVE ALGORITHM

In this algorithm the filter output and the error value are computed for each cycle. The error value is obtained by the difference of the desired output and the filter output. Obtained error value is used for updating the filter coefficient. The input equation X (k) is written as:

$$X(k) = \left[x(k), x(k-1).....x(k-K+1)\right]^T \quad (1)$$

Where *X(k)* is the input signal at time 'k' and 'T' is the transpose of the vector. The output signal *Y(k)* of an adaptive filter can be represent as

$$Y(k) = X^T(k)d(k) \quad (2)$$

Where d(k) is the filter coefficient vector and it is represented as

$$d(k) = \left[d_0(k), d_1(k), d_2(k), ...d_{(K-1)}(k)\right]^T \quad (3)$$

The Windrow Hoff LMS algorithm for filter coefficient updating can be represented as:

$$d(k+1) = d(k) + \mu e(k) X(k) \quad (4)$$

Where *e(k)* is the error signal, μ is the step size parameter which determines the convergence speed and accuracy of the filter. The error signal can be obtained by the difference between desired signal and the output signal. The equation is given as

$$e(k) = D(k) - Y(k) \quad (5)$$

Where *D (k)* is the desired signal. The input signal of the filter *X (k)* is nourishing into the delay line and then shifted to right for each sampling period.

Formulation of DA:

DA is a dynamic architecture for computing inner product vector with dynamic mechanism. Let us consider the inner product as:

$$y = \sum_{k=0}^{K} d_k x_k \quad (6)$$

Where d_k is fixed coefficient, x_k is the input signal and '*K'* is the number of input words. x_k is a 2's complement binary number scaled such that $|x_k| < 1$, then x_k can be expressed as:

$$x_k = -b_{k0} + \sum_{l=1}^{L-1} b_{kn} 2^{-l}$$

Where $$x_k = \{b_{k0}, \ldots\ldots b_{k(L-1)}\} \quad (7)$$

On substituting x_k in equation (6), we get the equation as follows:

$$y = \sum_{k=1}^{K} d_k \left[-b_{k0} + \sum_{l=1}^{L-1} b_{kl} 2^{-l} \right]$$

Above equation is the normal inner multiplication expression. By changing the summation order and rewriting the equation, finally we get equation as shown below which is the computation of distributed arithmetic.

$$y = \sum_{n=1}^{N-1} \left[\sum_{k=1}^{K} d_k b_{kn} \right] 2^{-n} + \sum_{k=1}^{K} d_k (-b_{k0})$$

3. PROPOSED MEMORY LESS DA (MLDA)

The proposed Memory-less DA (MLDA) filter architecture is explained in this section. It consists of 2:1 multiplexers instead of memory elements, and the adders of memory less DA architecture [13] are replaced with enhanced 4:2 compressor as shown in fig.2. MLDA consists of serial in parallel out shift register (SIPOSR), four 2:1 multiplexers, and enhanced 4:2 compressor adders. The input data x_k is fed to the SIPOSR and is given as one of the input of the 2:1 multiplexer and the other input of the multiplexer is the logic '0'. The selection lines for four 2:1 multiplexers are the filter coefficients. If the selection input line is high, then filter input will present at output else the output of multiplexer is zero. The outputs from multiplexers are given to the enhanced 4:2 compressor adder.

The enhanced 4:2 compressor adder is designed with dual mode logic (DML). DML logic consists of XOR/XNOR module and MUX module. The XOR/XNOR module is developed with CMOS logic and MUX module is developed using transmission logic gate (TG). By using DML realization, we can achieve better results in area and power. The A_1 and A_2 inputs are fed to XOR/XNOR1 module and A_3 and A_4 are fed to XOR/XNOR2 module. The outputs from XOR/XNOR modules are given to the MUX1 module with A_4 as a selection line. The outputs from MUX1 module are given as inputs to the MUX2 module with A_5 as selection line to generate sum. To achieve carry, A_4 and A_5 are given to MUX3 with selection line as one of the output of MUX2. Finally from MUX3 and MUX4 we get sum and carry output.

The proposed MLDA is implemented in Adaptive FIR filter. Arithmetic performance of higher order adaptive filters is so complex, so higher order filters are decomposed into smaller order adaptive FIR filter which makes the design implementation simple.

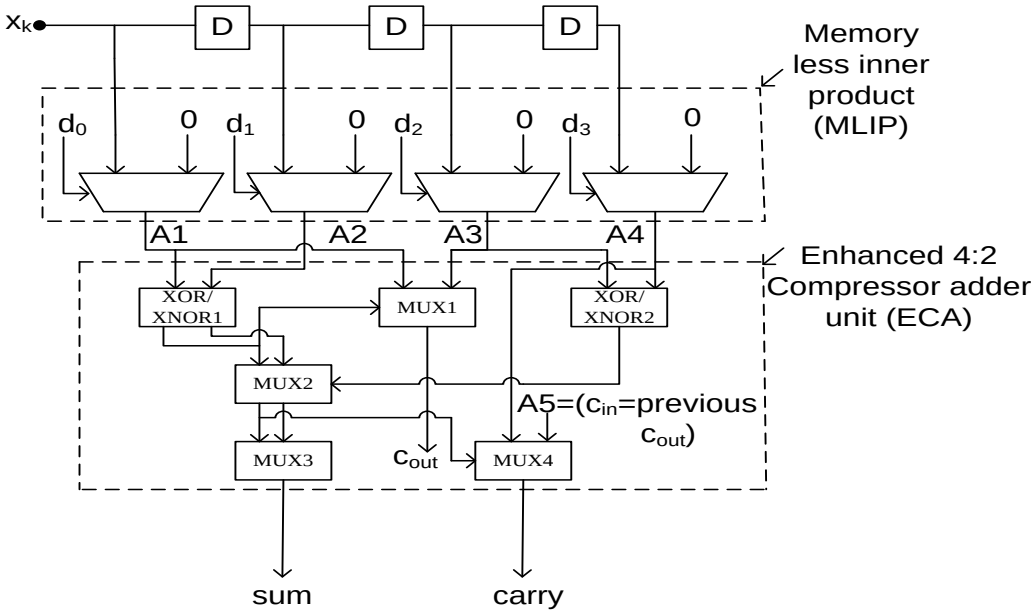


FIGURE 2. Proposed MLDA architecture

4. PROPOSED MLDA FOR SMALLER ORDER ADAPTIVE FIR FILTER

In this section, the proposed design is implemented for filter order K=4 and is shown in fig.3. The proposed design consists of Memory Less DA block (MLDA), weight increment block (WIB), sign-magnitude separator and control word generator block.

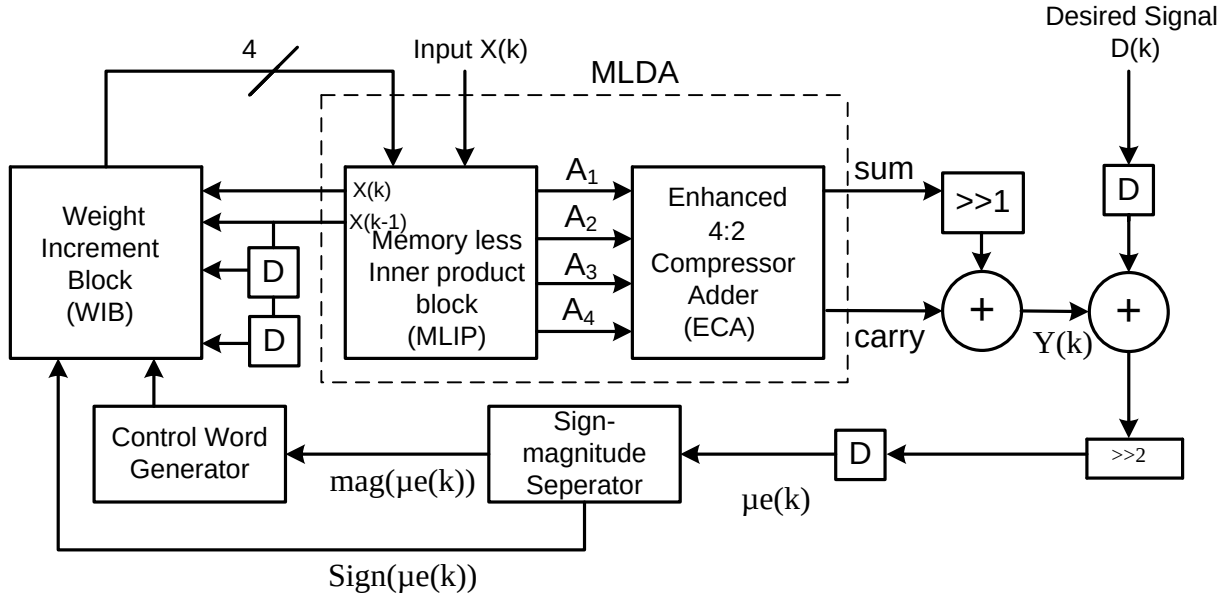


FIGURE 3. Four tap adaptive FIR filter using MLDA

The MLDA block consists of memory less inner product block (MLIP) and enhanced 4:2 compressor adder (ECA) block. The input X (k) is given to the MLIP block, the MLIP block consist of four 2:1 multiplexers, one input of 2:1 multiplexer is input signal and other input is logical '0' shown in fig.2. The selection lines for four multiplexers are the filter coefficients which are taken from the weight increment block. The four outputs from MLIP are A_1,

A_2, A_3, and A_4 are given to ECA. A_5 is the fifth input to the compressor which is the C_{out} of the previous stage compressor. The four inputs A_1, A_2, A_3, A_4 and sum output will have same weights and finally ECA obtain sum and carry. The generated sum and carry are added to obtain final filter output y(k). Obtained filter output y(k) is subsequently subtracted from the desired signal D(k) to achieve error sample e(k).

The input data X(k) is fed to the three D-ff and produces the signals of X(k-1),X(k-2) and X(k-3) signals which are given to the weight increment block (WIB). The block diagram of WIB is shown in fig.4. It consists of four barrel shifters, four adder/subtractor units, four delays and a word parallel bit serial converter. The barrel shifter shifts the different input values xk for k =0, 1, . . .,K − 1 by appropriate number of locations according to the control signal 't'. The barrel shifter obtains the required increments to be added with or subtracted from the present filter coefficient values. The sign bit of the error is used as the control for adder/subtractor unit. When sign bit is zero the barrel-shifter output is respectively added with the present filter coefficient else it is subtracted from the content of the corresponding present filter coefficient value in the weight register and are passed to word parallel bit serial converter. The output 'A' from WIB are given as selection lines for the four 2:1 multiplexer's of the memory less inner product block.

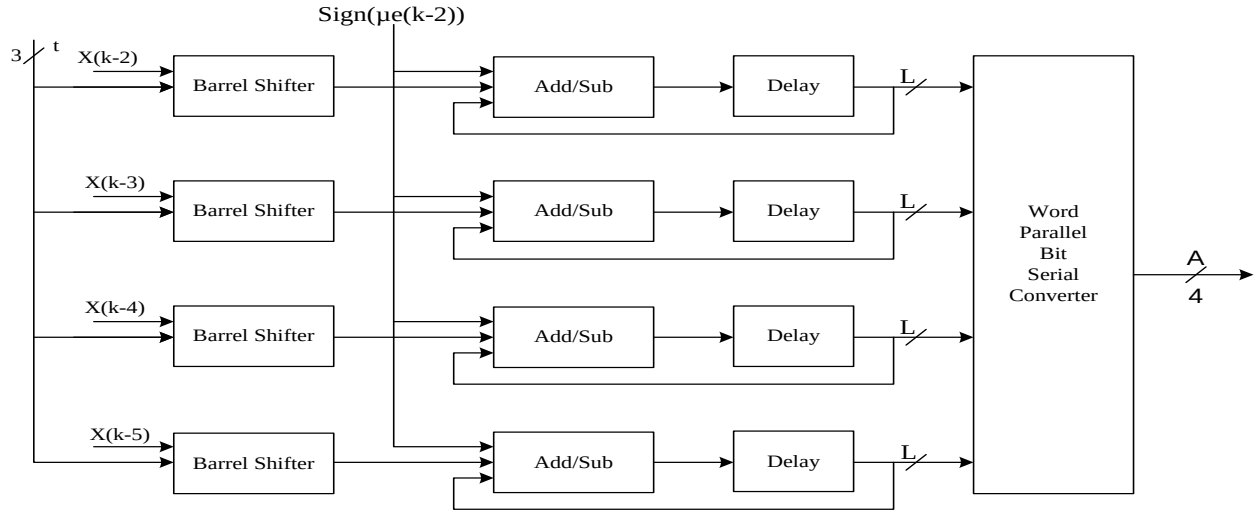


FIGURE 4. Weight Increment Block for k=4

5. PROPOSED MLDA FOR HIGHER ORDER ADAPTIVE FIR FILTER

The arithmetic performance of inner- product of equation (6) can be spitted into K/P. Let us consider K=PQ, where Q= 2^k , 'k' is the positive number and 'P' is the smaller adaptive filter block. Then the equation can be written as:

$$y = \sum_{k=0}^{P-1} d_k x_k + \sum_{k=P}^{2P-1} d_k x_k + ... \sum_{k=K-P}^{K-1} d_k x_k$$

Every P-point inner-product operation unit will perform according to the weight-increment unit to update P weights. The proposed architecture for K = 16 and P = 4 is shown in Fig. 5. It consists of four MLDA blocks, four WIB blocks, enhanced compressor adder, sign-magnitude separator and control word generator. The (L + 2)-bit 4 sums and 4 carries are produced by the four MLDA blocks and are added by the two 4:2 enhanced compressor adder and produces (L+4) bit 2 sums and 2 carries. (L+4) bit sums and carries are given to the enhanced 4:2 compressor to produce final sum and carry and get final output Y(k). Four carry-in bits should be added to sum words which are output of four 4-point inner-product blocks. Assuming that μ = 1/N, we truncate the four LSBs of $e(n)$ for K = 16 to make the word length of sign-magnitude separator be L bit. By using truncation, the performance of the adaptive FIR filter will not vary very much since the proposed architecture depends on the most significant one location of μe(k).

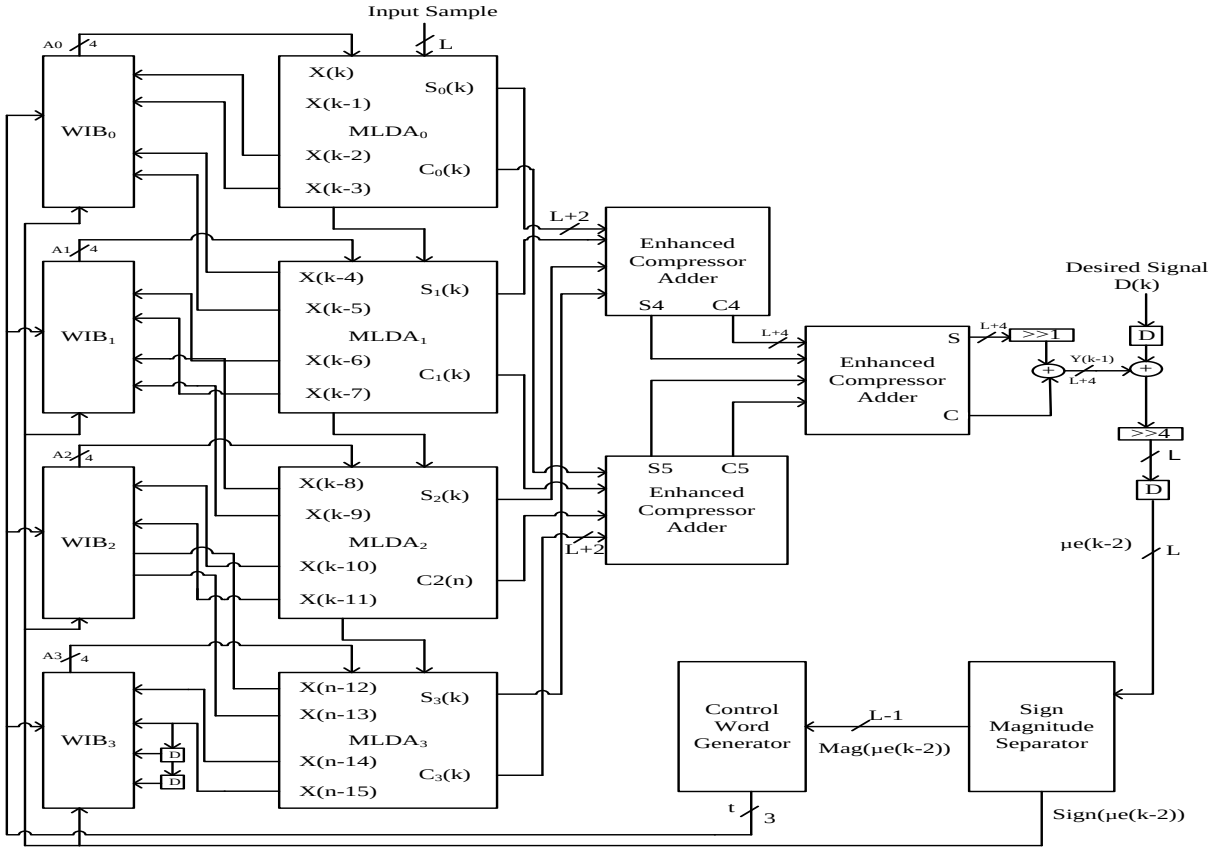


FIGURE 5. Adaptive FIR filter for filter length K=16

6. SYNTHESIS RESULT

Power, Area and timing have been estimated for the proposed design and comparative analysis is done with the existing architecture. The proposed design contains only four multiplexers in place of LUT, which is the main concern for area complexity. Multiplexers in the proposed design reduces the area significantly, moreover the compactness of the enhanced compressor adder gives an extra advantage to reduce the size of the filter more than 50%.The existing model and the proposed design has been coded in Verilog HDL and synthesized in Synopsys Design Compiler using 90nm SAED CMOS technology library for filter tap of K=16, 32 and 64 to compute the area, delay and power. From the synthesis result, it is clearly observed that the proposed design offers 52.79% less area and 36.50% less ADP when compared with pipelined DA based adaptive FIR filter [11]. 69.25% power reduction and 35% of PDP compared with the existing model and are tabulated in table 1.

Table 1: Synthesis Result Using SAED 90 nm Technology with L=8 and P=4

Design	Filter Length, K	MSP* (ns)	Throughput (per μs)	Area (sq. μm)	Power (mW)	ADP* (sq.um x ns)	PDP(mW X ns)
Park[11]	16	7.09	141.04	88719	2.917	629617.71	20.68153
	32	8.31	120.33	177460	5.834	1474692.60	48.48054
	64	9.6	104.167	364820	11.67	3502272.00	112.032
Proposed Design	16	11.8	84.745	41946	1.33	494962.8	15.694
	32	12.05	82.987	85262	1.695	1027407.00	20.42475
	64	13.11	76.278	170678	3.2548	2237588.58	42.6704

*MSP: Minimum sampling Period; ADP: Area Delay Product; PDP: Power Delay productThe layout diagram for the proposed MLDA based adaptive FIR filter is as shown in Fig.6.

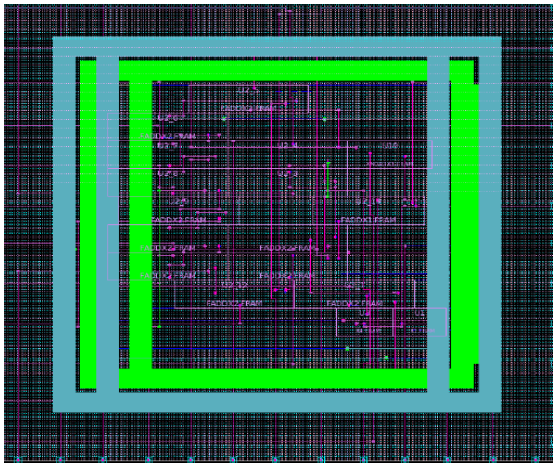


FIGURE 6. Layout chip diagram of Proposed MLDA based adaptive filter

Application of proposed method for Decision feed-back equalizer: In telecommunication, a signal transmitted undergoes distortions due to multi-path propagation and band limited channels which causes the inter symbol interference (ISI). ISI causes severe effect in wireless channel and makes the signal communication less reliable. Equalizer is a device that attempts to nullify the distortion occurred by a transmitted signal through a channel. Equalizers are placed at the receiver side to combat the ISI and to recover the transmitted signal. The classification of equalizers is detailed in Fig. 7. They are broadly classified as Linear equalizers and non-linear equalizers. Linear equalizers can eliminate the ISI, but enhances other noises which leads to poor signal performance. When the channel distortion is too severe and cannot be mitigated by linear equalizers, non-linear equalizers are used. The performance of non-linear equalizers is more effective to nullify the channel impairment. Decision feedback equalizer is the non-linear equalizer which is effectively used as channel equalizer. DFE gives better signal to noise ratio when compared with linear equalizer by removing ISI and it exhibits less noise. The basic architecture of DFE is shown in Fig. 8. It consists of a decision device (quantizer), a feed-forward (FF) filter, and a feed-back (FB) filter. The FF filter receives and equalizes the data with the transfer function of anti causal part of the channel and cancels pre-cursor ISI. The noise enhanced in DFE is significantly reduced. The FB filter suppresses the post-cursor ISI. The coefficients for the FF and FB filter should be carefully chosen to operate the DFE with zero errors. Until the quantizer propagates zero value, the DFE channel will operate efficiently with less noise.

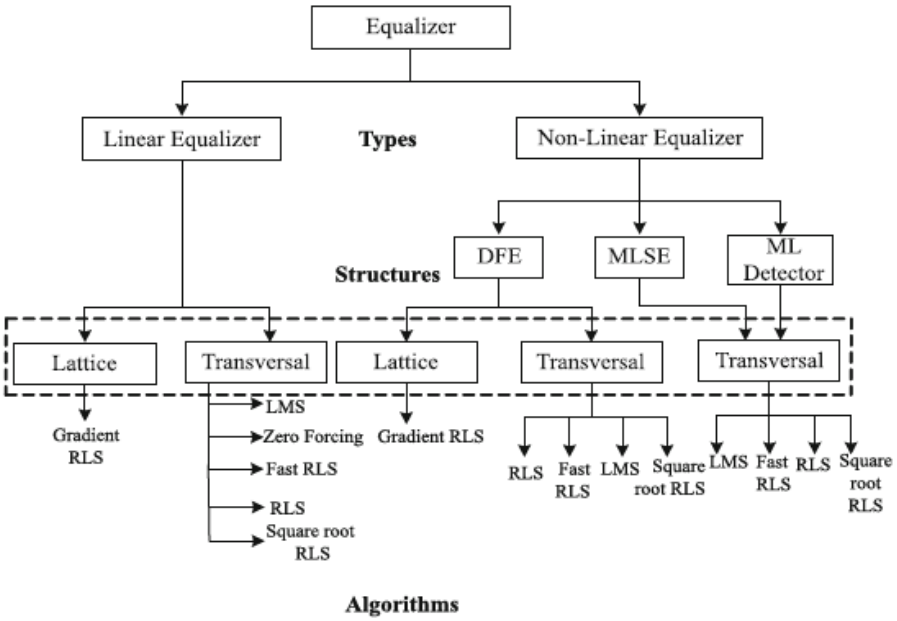


FIGURE 7. Flow chart of equalizer

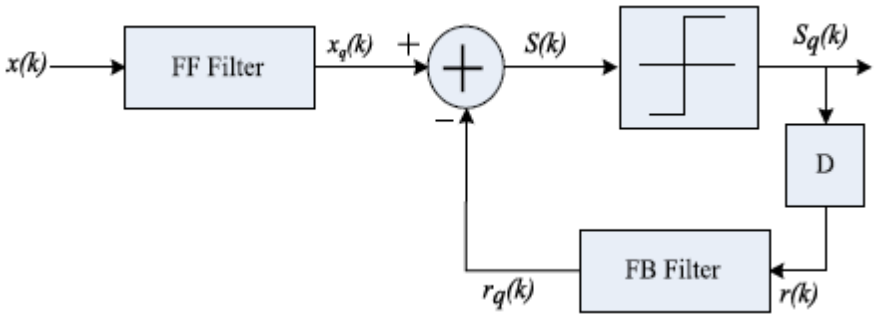


FIGURE 8. Block diagram of decision feedback equalizer

Let us consider, decision feed back equalizer shown in Fig. 8, with input signal *x(k)*, where $k \in Z$ with *Nf* number of FF filter coefficients and feedback output decision *r(k)* with *Nb* number of FB filter coefficients. The output generated decision *Sqk* for DFE is given as follows:

$$S_{qk} = Q[S(k)]$$
$$S(k) = x^{\wedge}(k) - r^{\wedge}(k)$$
$$r(k) = S_q(k-1)$$
$$r^{\wedge}(k) = \sum_{j=0}^{N_b-1} b_j r(k-j) = b^T r(k)$$

$$d^T = [d_0, d_1, d_3 ..... d_{N_f-1}]$$
$$x^{\wedge}(k) = \sum_{i=0}^{N_f-1} d_i x(k-i) = d^T x(k)$$
$$b^T = [b_0, b_1, b_2 ..... b_{N_b-1}]$$

The coefficient of FF filter is d^T and the coefficient of FB filter is b^T. N_b and N_f are number of FB and FF filter coefficients respectively. The 2's complementary form of x(k) and r(k) with 'W ' word length can be expressed as :

$$x(k-i) = -x_{i,W-1} + \sum_{w=1}^{W-1} x_{i,W-1-i} 2^{-i}$$

$$r(k-j) = -r_{j,W-1} + \sum_{w=1}^{W-1} r_{j,W-1-j} 2^{-j}$$

$$s(k) = \left[\sum_{i=0}^{N_f-1} -d_i x_{i,W-1} + \sum_{w=1}^{W-1} \sum_{i=0}^{N_f-1} x_{i,W-1-i} 2^{-i} \right]$$
$$- \left[\sum_{j=0}^{N_b-1} -b_j r_{j,W-1} + \sum_{w=1}^{W-1} \sum_{j=0}^{N_b-1} r_{j,W-1-j} 2^{-j} \right]$$

The difference of the outputs of FF and FB filter blocks is S(k) which is given to the decision device. The output S(k) is checked by the decision device whether the signal lies with in the range of signal or not and quantizes the signal according to the modulated scheme. The process will be continued until DFE gets zero error.

Testing of DFE using BPSK and FSK:

- The proposed DFE architecture with NMLDA has been tested by channel equalizer system with binary phase shift keying (BPSK) and frequency shift keying (FSK) modulated techniques.
- Let us consider that a set of channel impulse response signals are modulated with carrier signal using BPSK and FSK techniques and are transmitted.
- At receiving side of the channel equalizer, the received signal is generated with noise and ISI type of errors. For removing noise and ISI errors in the generated signal, the signals are passed to the DFE. In DFE, the FF filter block removes the precursor anti-causal part of ISI and the rest of the noise and errors are removed in the FB filter block.
- The DFE channel will operate until the decision device of the DFE propagates zero value. Finally, the original signal output is obtained with out errors in DFE.

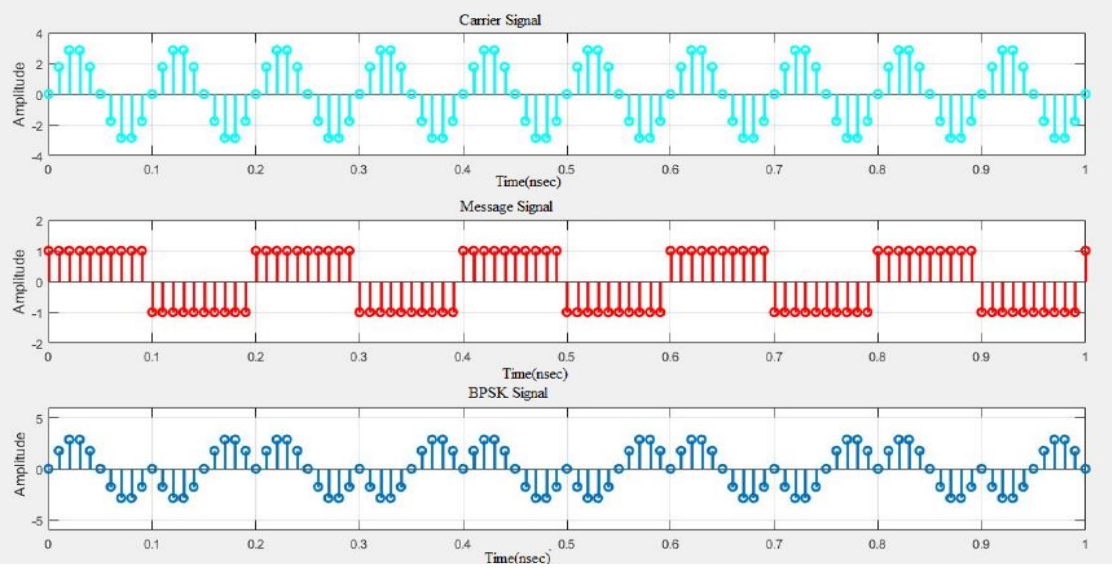


FIGURE 9. Testing of DFE using BPSK

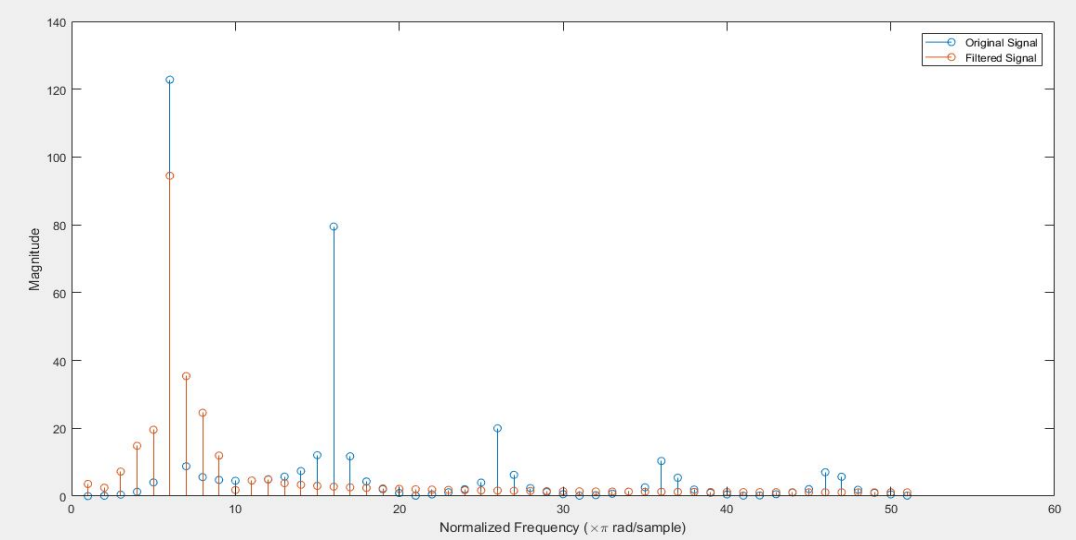


FIGURE 10. Original noise signal and filtered signal of DFE

7. CONCLUSION

In this paper, we have designed and implemented MLDA based adaptive FIR filter architecture for low power and area efficient. By using memory less DA the area of the adaptive filter is reduced. Area and power of the proposed design is further reduced by using enhanced 4:2 compressor adder. From the synthesis result we concluded that the proposed MLDA based design consumes 69.25% less power and 52.79% less area when compared with the pipelined DA based adaptive FIR filter. ADP and PDP of the proposed design is also less when compared with the existing pipelined DA based architecture.

REFERENCES

[1]. Thakor, Nitish V., and Y-S. Zhu. "Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection." IEEE transactions on biomedical engineering 38.8 (1991): 785-794.
[2]. Stenger, Alexander, Lutz Trautmann, and Rudolf Rabenstein. "Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filters." Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on. Vol. 2. IEEE, 1999.
[3]. Ihalainen, Tero, et al. "Channel equalization in filter bank based multicarrier modulation for wireless communications." EURASIP Journal on Advances in Signal Processing 2007.1 (2006): 049389.
[4]. Croisier, Alain, et al. "Digital filter for PCM encoded signals." U.S. Patent No. 3,777,130. 4 Dec. 1973.
[5]. Peled, Abraham, and Bede Liu. "A new approach to the realization of nonrecursive digital filters." IEEE Transactions on Audio and Electroacoustics 21.6 (1973): 477-484.
[6]. Zohar, Shalhav. "A realization of the RAM digital filter." IEEE Transactions on Computers 25.10 (1976): 1048-1052
[7]. White, Stanley A. "Applications of distributed arithmetic to digital signal processing: A tutorial review." IEEE Assp Magazine 6.3 (1989): 4-19.
[8]. White, Stanley A. "On mechanization of vector multiplication." Proceedings of the IEEE 63.4 (1975): 730-731.
[9]. Allred, Daniel J., et al. "LMS adaptive filters using distributed arithmetic for high throughput." IEEE Transactions on Circuits and Systems I: Regular Papers 52.7 (2005): 1327-1337.
[10]. Guo, Rui, and Linda S. DeBrunner. "Two high-performance adaptive filter implementation schemes using distributed arithmetic." IEEE Transactions on Circuits and Systems II: Express Briefs 58.9 (2011): 600-604.
[11]. Park, Sang Yoon, and Pramod Kumar Meher. "Low-power, high-throughput, and low-area adaptive FIR filter based on distributed arithmetic." IEEE Transactions on Circuits and Systems II: Express Briefs 60.6 (2013): 346-350.
[12]. Mohanty, Basant K., Pramod Kumar Meher, and Sujit K. Patel. "LUT optimization for distributed arithmetic-based block least mean square adaptive filter." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 24.5 (2016): 1926-1935.
[13]. Ghamkhari, Seyedeh Fatemeh, and Mohammad Bagher Ghaznavi-Ghoushchi. "A new low-power architecture design for distributed arithmetic unit in FIR filters implementation." Circuits, Systems, and Signal Processing 33.4 (2014): 1245-1259.

