International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-10, August 2019

Low Power High Throughput Memory Less Adaptive Filter using Distributed Arithmetic

Paida Chiranjeevi, B. Venkatesu

Abstract: This paper briefs an area efficient, low power and high throughput LMS adaptive filter using Distributed Arithmetic architecture. The throughput is increased because of parallel updating of filter coefficient and computing the inner product simultaneously. Here we have proposed memory-less design of distributed arithmetic (MLDA) unit. The proposed design uses 2:1 multiplexer's architecture to replace LUT of the conventional DA to reduce the overall area of the filter. Enhanced compressor adder is used for accumulation of the partial products, which further helps to reduce the area. Parallel updating of the generation and accumulation enhance the throughput of the design. The proposed architecture requires more than half area that required for the existing LUT based inner product block. The proposed design is implemented in synopsis design compiler and the result shows that the area decreased by 52.7% and also the MUX based DA for the 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).

Key words: Adaptive Filter; Distributed Arithmetic; Inner Product Block; weight-increment

I. INTRODUCTION

Adaptive Filter is a linear model where the filter coefficients are variable unlike other filters. In this the filter coefficients are revised for every iteration so that the error between the accurate signal and desired signal is less. The multipliers and adders take more computation time and occupy more area hence MAC blocks of filters are replaced by multiplier less architectures like Distributed Arithmetic. Distributed Arithmetic (DA) is chosen for implementing because it takes less computing time and area efficient [1,4,5,6,7]. It is mainly used for calculating Sum of Product and it also saves a lot of area in DSP application. It is widely used for various applications such as telecommunication, audio and video signal processing, and noise and echo cancellation. Adaptive filter design can be analog, digital or can be mixed type with respect to its advantages and disadvantages. Adaptive Filters can also be of different types. It can be linear or non-linear, finite or infinite impulse response can also have single input or multi input. In brief of above discussion we can be as follows.

1) Due to parallel Multiplexer based design the throughput increases rapidly.
2) Further increase in throughput is done by simultaneously incrementing in weight update block.
3) Accumulation which is done by conventional adder is replaced by enhanced compression adder for less area and reduction of time complexity of the design.

Manuscript published on 30 August 2019.
*****Correspondence Author(s)
Paida Chiranjeevi, Electronics and communication engineering, Shree College, Gopalapuram, Andhra Pradesh ,India
B.Venkatesh, Electronics and communication engineering, Shree College, Gopalapuram, Andhra Pradesh, India.

In this design of the filter, there are four major blocks. 4-point IPB, WIB block, sign-magnitude separator and control word generator. IPB computes the partial product for the filter and the WIB dynamically updates the filter coefficients. Forward path becomes the critical path because of inner-product block which results in the desired output. Thus, high sampling frequency cannot be used for input signal as the critical path increases and ultimately becomes greater than sampling rate.

The layout of the paper is as follows. Section 2 describes the background of LMS adaptive algorithm. Section.3 explains about the proposed design. Result analysis is explained in section.4. Finally conclusion is explained in section.5.

II. BACKGROUND OF ADAPTIVE ALGORITHM:

In this algorithm the output of the filter and the error i.e., the difference between the desired output and the filter output is computed for each cycle which is equal to. The equation for weight updating of an nth iteration is :

$$d(n+1) = d(n) + \mu \cdot e(n) \cdot x(n) \qquad (1a)$$

Error e(n) and output of filter y(n) can be written as,

$$e(n) = D(n) - y(n) \qquad (1b)$$
$$y(n) = d_q^T(n) \cdot x(n) \qquad (1c)$$

Where
d(n) = weight vector,
D(n)= desired response,
μ = factor of convergence
x(n)= input vector and

$$x(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^T \qquad (2a)$$

$$w(n) = [w_0(n), w_1(n), \ldots, w_{N-1}(n)]^T \qquad (2b)$$

For pipelined structure, error signal is achieved after certain number of cycles, which is termed as "adaptation delay". 'e(n-m)'- delayed error is used in updating the weight-increment block where m be the adaptation delay of the filter. Weight update equation for delayed Least Mean Square adaptive filter is given by:

$$d(n+1) = d(n) + \mu\ e(n-m)\ x(n-m) \qquad (3)$$

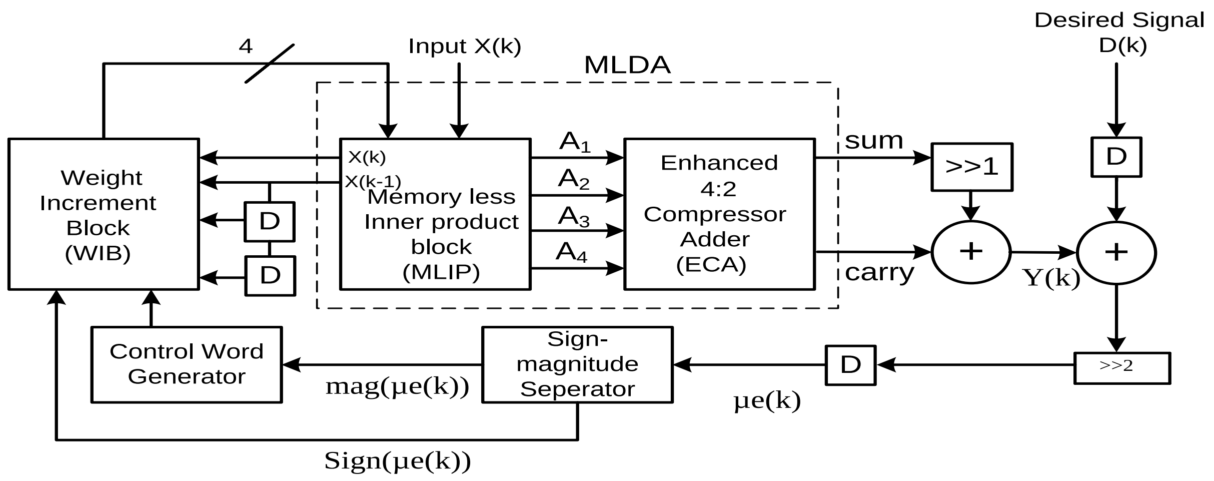

Fig. 1: Proposed Least Mean Square

Adaptive filter of length (N) 4

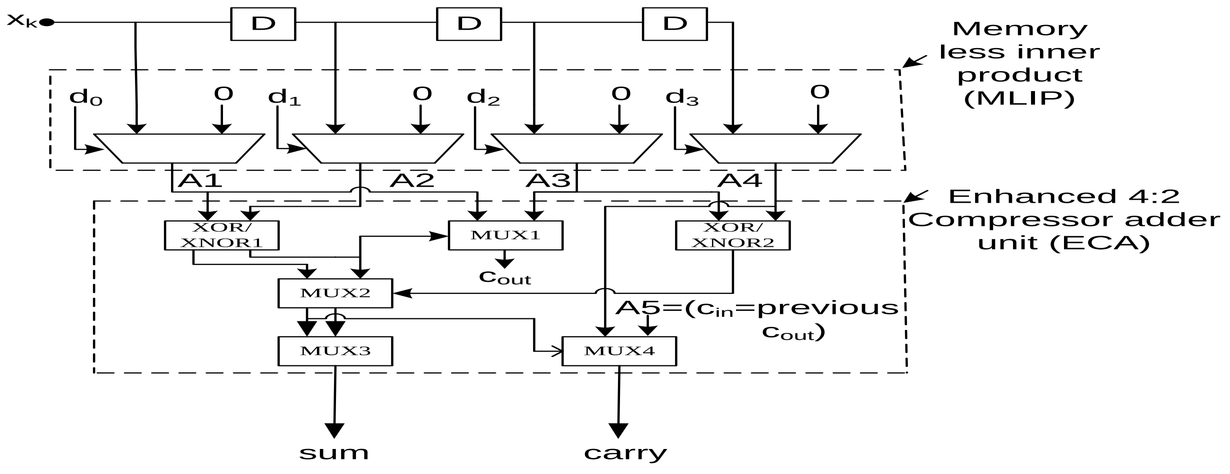

Fig.2 Proposed memory less DA

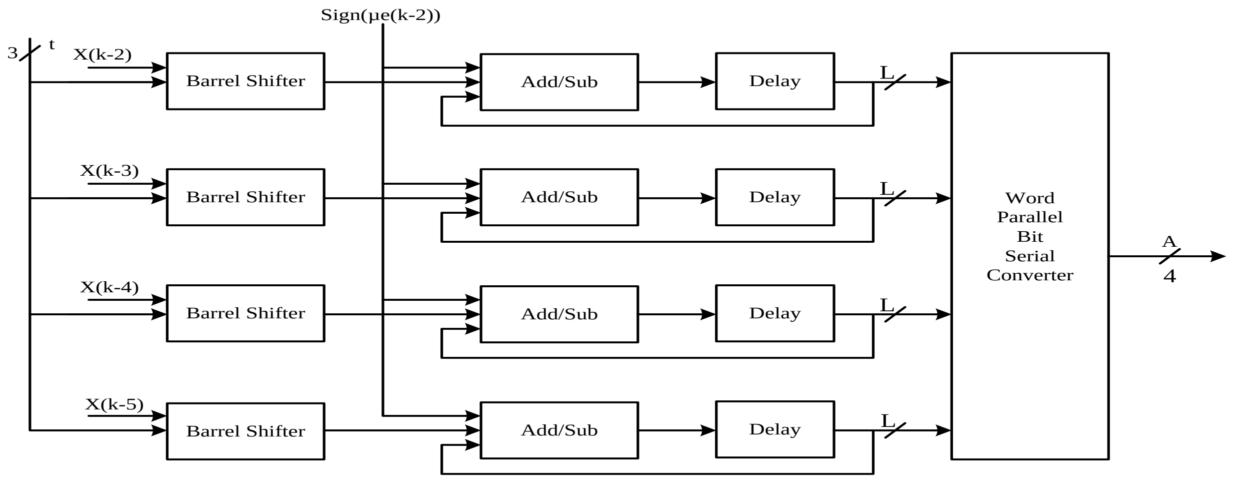

Fig3: Weight Increment Block for filter of length N=4

III. PROPOSED MLDA BASED ADAPTIVE FILTER:

Fig.1 shows the proposed MLDA based adaptive FIR filter. It consists of weight increment block, memory less inner product block and the compressor adder unit.

Since the LUT design consumes larger area for inner product computation hence it is replaced by multiplexers for area efficiency and is shown in fig.2. The adaptive weights are connected to the selection line. These determine input of the multiplexers. Depending on the selection line filter inputs will be passed to the accumulator in a product form with the weights coming from the WIB of the adaptive filter. As shown in Fig.3, the selection lines {d0 d1 d2 d3} are updated with the filter weights [8].

The inner product is calculated by shift accumulation in L cycles. On the Lth cycle first valid partial sum will be read for the corresponding L bit input sequences $\{d_{kl}\}$ for $0 \le l \le L-1$. The conventional shift adder is replaced with enhanced compressor adder to further reduce area. In the proposed design one enhanced compressor adder is sufficient to compute the inner product instead of three conventional adders for the same purpose.

The computation of inner- product in equation (2) be broken into in the form of K/P (assume that K=PQ, where Q is considered to be 2^n, and n be the positive integer) small adaptive filtering blocks of filter length P as:

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

The filter coefficients are updated by the corresponding weight increment block. The updated coefficient values are connected to the selection line of multiplexers present in the inner product block. Fig.4. shows 4-point IP blocks each of P=4.

The (L+2) bit sum of products of the filter coefficients and input vectors are coming out of the four blocks of IP blocks. The four outputs of the four inner product blocks are summed up together using enhanced compressor adder and the sum thus obtained is compared with the desired input value. The error, e(n) (difference between actual result and the desired result) is then treated for generating the updated filter coefficients. Since μ is of the order of 1/N i.e., μ=O(1/N), it can be assumed that μ =1/N. for the filter tap N=16, we truncate the four LSB of e(n) to make the word length to be sign-magnitude to be L bit. Performance of the filter does not get affected much, since the truncation is happened from the LSB.

IV. SYNTHESIS RESULTS:

Power, Area and timing have been estimated for the proposed design and comparative analysis is done with the existing architecture. Removal of the LUT reduce 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%. Proposed designs are synthesized for tap N=16, 32 and 64 to compute the area, delay and power. After analysing the synthesis result, it is clearly observed that proposed architectures offer 52.79%, 50.5% lesser area and 36.50%, 32.435% less ADP. 69.25%, 76.297% power reduction respectively for proposed design 1 & 2 compared with the existing LUT based model. By using parallel inner product computation and the shift accumulation operation the throughput rate is significantly raised as compared to the conventional design of DA [3].

V. CONCLUSION:

In this paper an area efficient pipelined architecture is developed. Simultaneous operation of inner product computation and the shift accumulation improves the performance of the DA unit, which in turn increase the throughput of the Adaptive filter. The problem of large area consumption of the LUT is resolved by the use of 2:1 MUX for inner product computation. Along with this, the enhanced compressor adder used in the proposed design also reduces the overall filter area. From the synthesis result we can note that the proposed MUX based design consumes 3.25 times less power and 1.57 times less area.

REFERENCES:

1. “Low-Power, High-Throughput, and Low-Area Adaptive FIR Filter Based on Distributed Arithmetic” Sang Yoon Park, Member, IEEE, and Pramod Kumar Meher, Senior Member, IEEE
2. “Applications of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review, Stanley A. White, IEEE ASSP Magazine, July, 1989”
3.]D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, “LMS adaptive filters using distributed arithmetic for high throughput,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 7, pp. 1327–1337, Jul. 2005.
4. NagaJyothi, Grande, and Sriadibhatla SriDevi. "Distributed arithmetic architectures for fir filters-a comparative review." 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). IEEE, 2017
5. Grande, Naga Jyothi, and Sriadibhatla Sridevi. "Asic implementation of shared lut based distributed arithmetic in fir filter." 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS). IEEE, 2017.
6. Jyothi, Grande Naga, and Sridevi Sriadibhatla. "Asic implementation of low power, area efficient adaptive fir filter using pipelined da." Microelectronics, Electromagnetics and Telecommunications. Springer, Singapore, 2019. 385-394.
7. NagaJyothi, Grande, and Sriadibhatla Sridevi. "High speed and low area decision feed-back equalizer with novel memory less distributed arithmetic filter." Multimedia Tools and Applications (2019): 1-15.
8. Jyothi, Grande Naga, and Sriadibhatla Sridevi. "Low Power, Low Area Adaptive Finite Impulse Response Filter Based on Memory Less Distributed Arithmetic." Journal of Computational and Theoretical Nanoscience 15.6-7 (2018): 2003-2008.

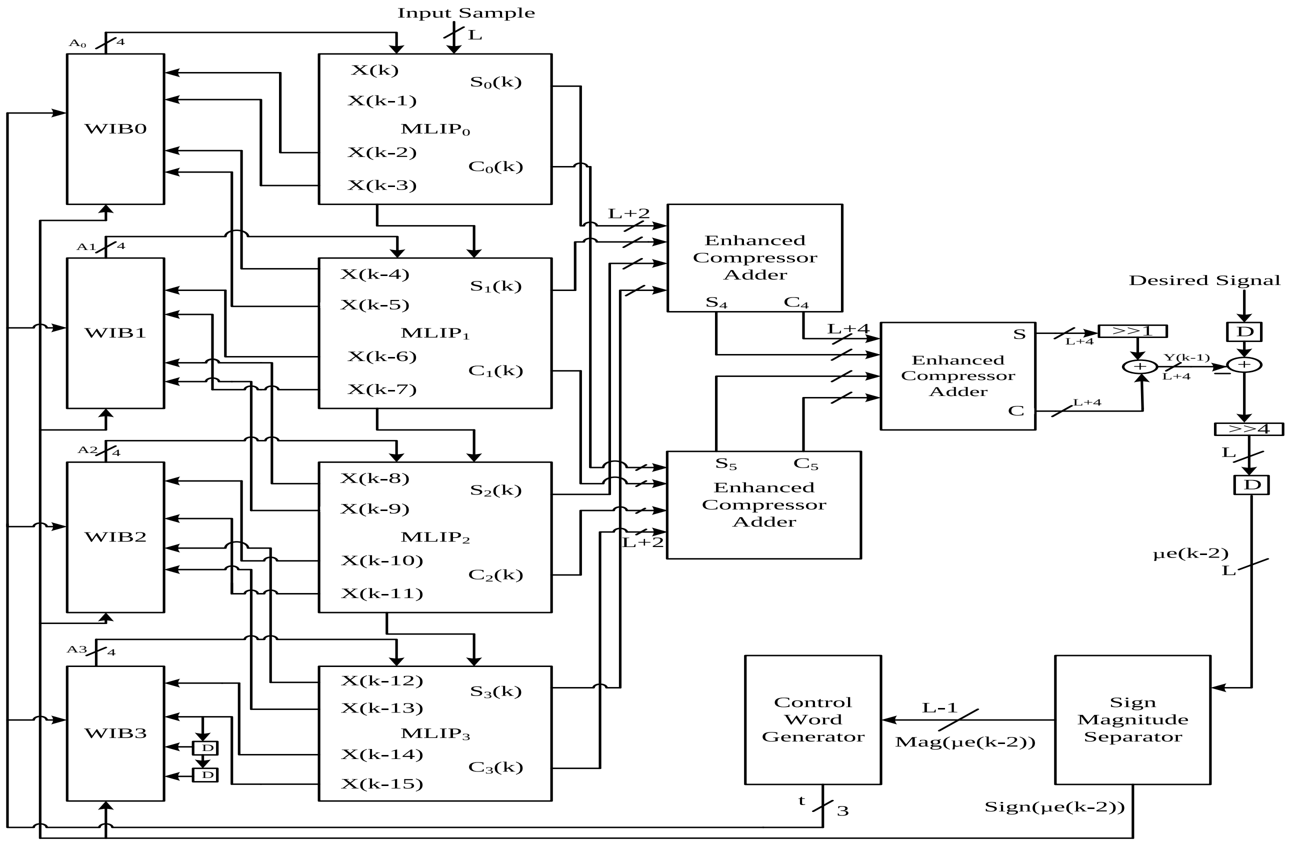

Fig. 4: LMS Adaptive Filter of length N=16 and P=4 based on DA

*IPB- Inner Product Block; WIB- Weight Increment Block

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

Model	Size of Filter, N	DAT* (ns)	Throughput (per µs)	Area (sq. µm)	Power (mW)	ADP* (sq.um x ns)
LUT Based Design	16	7.09	141.04	88719	2.917	629617.71
	32	8.31	120.33	177460	5.834	1035301.64
	64	9.6	104.167	364820	11.67	4256719.76
Proposed Design	16	11.8	84.745	41946	1.33	494962.8
	32	12.05	82.987	85262	1.695	1027407.1
	64	13.11	76.278	170678	3.2548	2237588.58

*DAT: Data Arrival Time; ADP: Area Delay Product