Journal of Engineering and Applied Sciences 14 (20): 7756-7764, 2019
ISSN: 1816-949X

Design and Implementation of LMS Adaptive FIR Filter Based on OBC Distributed Arithmetic Algorithm with MCSLA

[1]D. Kalaiyarasi and [2]T. Kalpalatha Reddy
[1]Department of Electrical and Electronics Engineering,
Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India
[2]Department of Electronics and Communication Engineering,
Sri Venkateswara Engineering College for Women, Tirupati, Andhra Pradesh, India
kalaiccarthi@yahoo.co.in, 9566119388

Abstract: Distributed Arithmetic (DA) is a methodology used to save resources in MACs implementing DSP functions. Without using multipliers, DA has been computing the multiply-accumulate process. In conventional DA, the partial products of the filter coefficients have been pre-computed and stored in parallel Lookup Tables (LUTs). The results of parallel LUTs have been given to Carry Select Adder (CSLA) with the help of multiplexer. The sizes of the hardware multiply accumulate has been reduced in DA based adaptive filter which is well suited to FPGA design. It also reduces the number of logic elements in the design. In the proposed method, OBC based DA in LMS adaptive filter has been implemented to reduce the half of the logic elements when compared to the conventional DA. Offset Binary Coding (OBC) is a digital coding scheme where all-zero corresponds to the minimal negative value and all-one to the maximal positive value. Nothing but logic elements but also, the power and time have been reduced over the conventional method. The main concern of the proposed method is the throughput. Throughput is defined as the total clock rates by the number of clock cycles needed for filtering and updating of filter coefficients. The proposed architecture has been implemented in Quartus II 9.1 sp 1 Web Edition with the device as Stratix-EP2S15F484C3. The proposed architecture for length N = 16 achieves 14.4% in DATs, 44.26% in No. of logic elements, 46.57% in No. of registers and 2.25% in power. Likewise, the proposed architecture for length N = 32 offers 4.28% in DATs, 70.6% in No. of logic elements, 71.93% in No. of registers and 6.5% reduction in power consumption.

Key words: Least Mean Square (LMS) adaptive filter, Distributed Arithmetic (DA), Modified Carry Select Adder (MCSLA), Shift accumulate, Offset Binary Coding (OBC), Lookup Tables (LUTs)

INTRODUCTION

The linear filtering such as channel equalization, echo cancellation, system identification, signal de-noising, etc. have been adapted to changes in the input signals. Adaptive FIR filters have been widely used in DSP applications. The filter and error computation and weight updating are the two blocks in an adaptive FIR filter. Least Mean Square (LMS) adaptive filter has been one of the efficient filters used to find out the filter coefficients. Due to its simplicity and complexity, the LMS adaptive filter has been used in many signal applications (Haykin and Widrow, 2003). The famous algorithm called Widrow-Hoff Least Mean Square filter has to update the tapped delay line of FIR filter weights.

If the input signal has a high sampling rate, then it is necessary to reduce the critical path. To resolve this issue, multiply and accumulate operation in the direct form structure has been replaced by a series of Lookup Tables (LUTs) and shift accumulator. Hence, the replacement by a series of lookup tables and shift accumulator are known as Distributed Arithmetic (DA) (White, 1989). In the DA technique, the multiplication and accumulate gives a moderate increase in the usage of memory. DA with LUT provides better efficiency in FPGA instead of multiply and accumulate operation (Allred *et al*., 2004).

DA technique has more popularity because of its throughput processing, low power, low area and its cost. Instead of using multiple and accumulate operation, the DA technique achieves high throughput, low area and low power over DSP microprocessor architectures (Allred *et al*., 2005). One of the LMS filters named Block LMS (BLMS) filter has been implemented by DA techniques. Only one set of LUTs has been modified in

Corresponding Author: D. Kalaiyarasi, Department of Electrical and Electronics Engineering,
Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India, kalaiccarthi@yahoo.co.in,
9566119388

every iteration from M sets in LUT based weight updating scheme for BLMS algorithm. By using BLMS adaptive digital filter, LUT access per output has been achieved efficiently (Guo and DeBrunner, 2011a).

The Carry Save Accumulator (CSA) in the adaptive filter achieves better area reduction, power consumption and time complexity (Joanna and Anathalakshmi, 2015). The main aim of digital filters is Distributed Arithmetic (DA) which performed to design bit-level architectures for vector-vector multiplication for the implementation of convolution (Guo and DeBrunner, 2011b). The offset binary coding scheme based DA has been updated to LUTs from time to time. It consumes less chip area and high throughput (Chandekar and Pawar, 2017).

For example, a 128-tap finite-impulse-response adaptive filter with the proposed implementation produces 12 times more throughput (for k = 8) and consumes almost 26% less area when compared to the best of existing architectures (Prakash and Shaik, 2013). An adaptive FIR filter with block least means square has been mainly used for the reduction of power consumption. In advance, the distributed arithmetic based block least mean square adaptive filter achieves better area reduction and also in power (Mohanty *et al*., 2015). An efficient Distributed Arithmetic (DA) provides a high-throughput of Finite Impulse Response (FIR) filters whose filter coefficients change during runtime (Park and Meher, 2014).

Literature review

LMS adaptive filter algorithm: In DSP digital filters, one of the efficient filters is the LMS adaptive filter which predicts one random process y(n) from the observation of input random process. LMS adaptive filters have been mainly used to detect the filter coefficients from the input and output signals. Error minimization has been mainly resolved by Least Mean Square (LMS) algorithm due to its simplicity, stability and fast convergences. In every iteration, the LMS algorithm computes a filter output and an error. The barrel shifter is used to shift the data by a specified number of bits without the use of sequential logic and the shift distance is used to execute the sequence of multiplexers. Let x(n) and h(n) are the input sample vector and filter coefficient vector at nth iteration for N-tap filter has been given as:

$$x^T(n)=\left[x(n),x(n\text{-}1),x(n\text{-}2),...,x(n\text{-}N\text{+}1)\right] \qquad (1)$$

$$h^T(n)=\left[h_0(n),h_1(n),h_2(n),...,h_{N\text{-}1}(n)\right] \qquad (2)$$

During nth iteration, the filter output is given by:

$$y(n)=h^T(n).x(n) \qquad (3)$$

and the estimated error denoted by e(n) is given by:

$$e(n)=d(n)\text{-}y(n) \qquad (4)$$

where, d(n) is the desired signal. The updation of filter coefficient vector at (n+1)th iteration is given by:

$$h(n+1)=h(n)+\mu e(n)x(n) \qquad (5)$$

where, μ is the convergence factor.

Distributed Arithmetic (DA): Distributed Arithmetic is used to compute multiply-accumulate operation without using multipliers. It converts multiply-accumulate operation into shift accumulation. It is a multiplierless technique used for reducing the sizes of the hardware multiply accumulate that is well suited to FPGA design. At nth iteration, the output y(n) of a N-tap FIR filter is given by:

$$y(n)=\sum_{i=0}^{N\text{-}1}h(i)x(n\text{-}i) \qquad (6)$$

In simplified form:

$$y=\sum_{j=1}^{N-1}h_i x_i \qquad (7)$$

where, h_i and x_i for $0 \leq i \leq N\text{-}1$ forms N-point vectors corresponding to the current filter coefficients and most of the input samples. The input samples have been expressed in k-bits 2's complement representation:

$$x_i=\text{-}x_{i0}+\sum_{j=1}^{k-1}x_{ij}2^{\text{-}j} \qquad (8)$$

where, x_{ij} denotes jth bit of x_i. Substituting (8), Eq. 7 has to be extended:

$$y=-\sum_{i=0}^{N-1}h_i x_{i0}+\sum_{i=0}^{N-1}[\sum_{j=1}^{k-1}h_i x_{ij}2^{-j} \qquad (9)$$

The conversion of sum of product forms into distributed form, the order of the summation over indices i and j. Hence, it has been interchanged into:

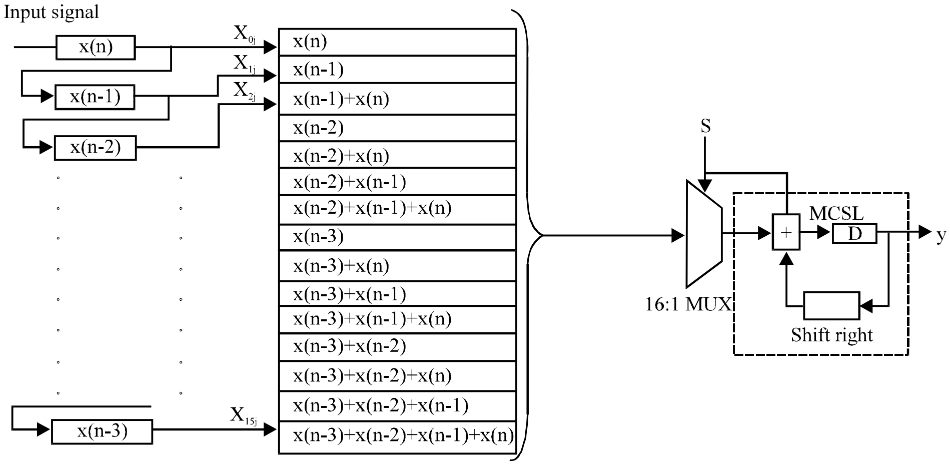

Fig. 1: DA based N-tap FIR filter

$$y = -\sum_{i=0}^{N-1} h_i x_{i0} + \sum_{j=1}^{k-1}\left[\sum_{i=0}^{N-1} h_i x_{ij} 2^{-j}\right] \tag{10}$$

$$y = \sum_{i=0}^{k-1} h_j 2^{-j} \text{ where } h_j = \sum_{i=0}^{N-1} h_i x_{ij} \tag{11}$$

The implementation of DA based N-tap FIR filter has been shown in Fig. 1.

MATERIALS AND METHODS

Proposed OBC based DA architecture: Distributed Arithmetic (DA) architecture is a bit serial operation used to compute multiply-accumulate operation without multipliers. The numbers of input addresses have been given to ROM with 2^N words. ROM is a read-only memory act as a LUT in the DA architecture. For 2^N words in ROM, the operation takes place according to, the input address given to LUT. Here, the multiply-accumulate operation is converted to a shift accumulator where the Modified Carry Select Adder (MCSLA) is used for the addition operation with the shifter as feedback. It is multiplierless techniques used for reducing the hardware multiply accumulate which is well suited to FPGA design (Fig 2).

One of the advanced techniques used over DA architecture is Offset Binary Coding (OBC) scheme. OBC is also termed as excess-K, excess-N, excess code or biased representation. It is a digital coding scheme where all zero corresponds to the minimal negative value and all one to the maximal positive value. The offset K for an n-bit binary word K = 2^{N-1}:

$$x(n-k) = 1/2\left[x(n-k) - \left[-\left(x(n-k)\right)\right]\right] \tag{12}$$

$$-x(n-i) = -\overline{x_{i0}} + \sum_{j=1}^{k-1} \bar{x}_{ij} 2^{-j} + 2^{-(N-1)} \tag{13}$$

Substitute, Eq. (8) and (13-12):

$$x(n-i) = 1/2\left[-xi0 + \sum_{j=1}^{k-1} x_{ij} 2^{-j} - \overline{x_{i0}} + \sum_{j=0}^{k-1} x_{ij} 2^{-j} + 2^{-(k-1)}\right] \tag{14}$$

Bu defining D_{ij} as . $x_{ij} 2^{-j} - \overline{x_{i0}}$, the output from FIR filter can be written as:

$$y = -\sum_{i=0}^{N-1} \frac{h_i D_{i0}}{2} + \sum_{j=1}^{k-1}\left[\sum_{i=0}^{N} \frac{w_k D_{ij}}{2}\right] 2^{-j} - \sum_{i=0}^{N-1} \frac{h_i}{2} 2^{-(i-1)} \tag{15}$$

By defining E_j as $\sum_{i=0}^{N-1} \frac{h_i D_{ij}}{2}$ and E_{extra} as $-\sum_{i=0}^{N-1} \frac{h_i}{2}$:

Modified Carry Select Adder (MCSLA) is the advanced one of normalized Carry Select Adder (CSLA). When compared to CSLA, MCSLA have less number of gate count values. This MCSLA is used in a shift accumulator to reduce the total number of areas. The required Modified Carry Select Adder (MCSLA) used in the proposed OBC based DA table for the 4-tap FIR filter is shown in Fig. 3. Least number of logic gates has been used to design Reduced Full Adder (RFA). The performance of digital adder circuits has been ease by multiplexer based RFA circuit. Gate count of full adder is resolute as follows:

- Gate count of FA = Gate count [(2*XOR)+(2*AND)+(1*OR)]
- Gate count of FA = [(2*5)+(2*1)+(1*1)] = 10+2+1 = 13

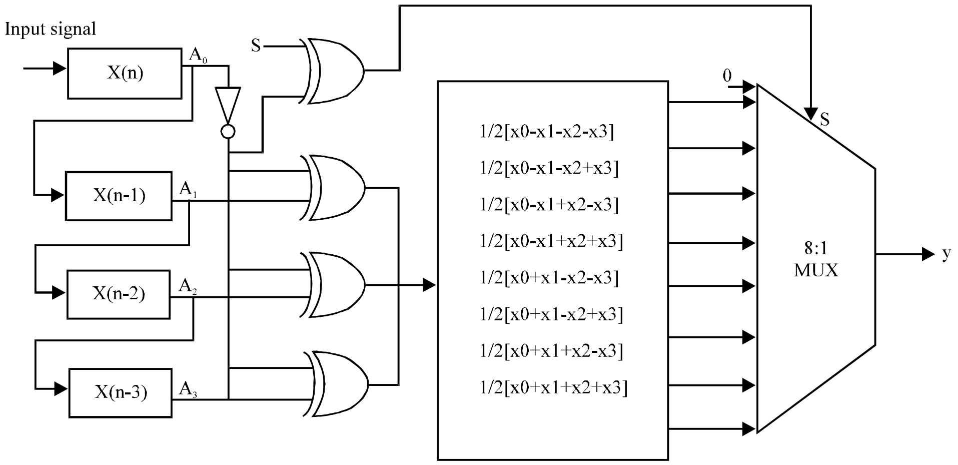

Fig. 2: Proposed OBC based DA table for 4-tap FIR filter

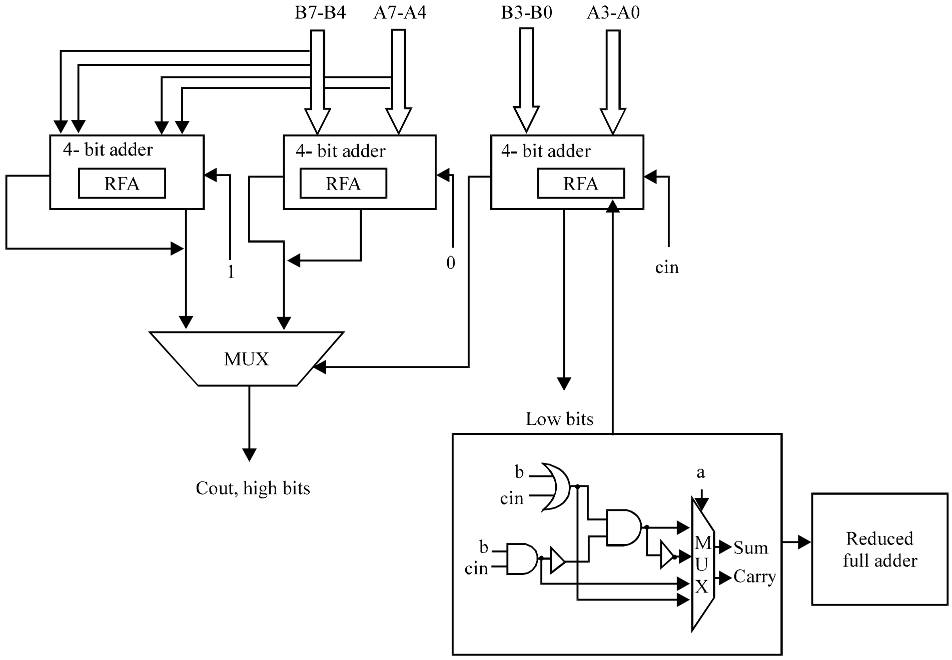

Fig. 3: Modified carry select adder

- Gate count of reduced full adder = Gate count [(2*AND)+(1*OR)+(2*NOT)+(1*MUX)]
- Gate count of Reduced full adder = [(2*1)+(1*1)+(2*1)+(1*4)] = 2+1+2+4 = 9

RFA design is simplified by using of De Morgan's Theorem and Boolean logic. General expression to find the sum and carry of RFA is given in Eq. 16 and 17:

$$\text{Sum} = \sum_{A=0}^{1}\left[X\overline{A}+\overline{X}A\right] \tag{16}$$

Where:

$$X = (B+C).\,\overline{BC}$$
$$= B\overline{C}+C\overline{B}$$
$$= B \oplus C$$
$$\overline{X} = \overline{(B+C).\overline{BC}}$$
$$= \overline{\left(B\overline{C}+C\overline{B}\right)}$$
$$= BC+\overline{BC}$$
$$= \overline{B \oplus C}$$

$$\text{Carry} = \sum_{A=0}^{1}\left[BC\overline{A}+(B+C)A\right] \tag{17}$$

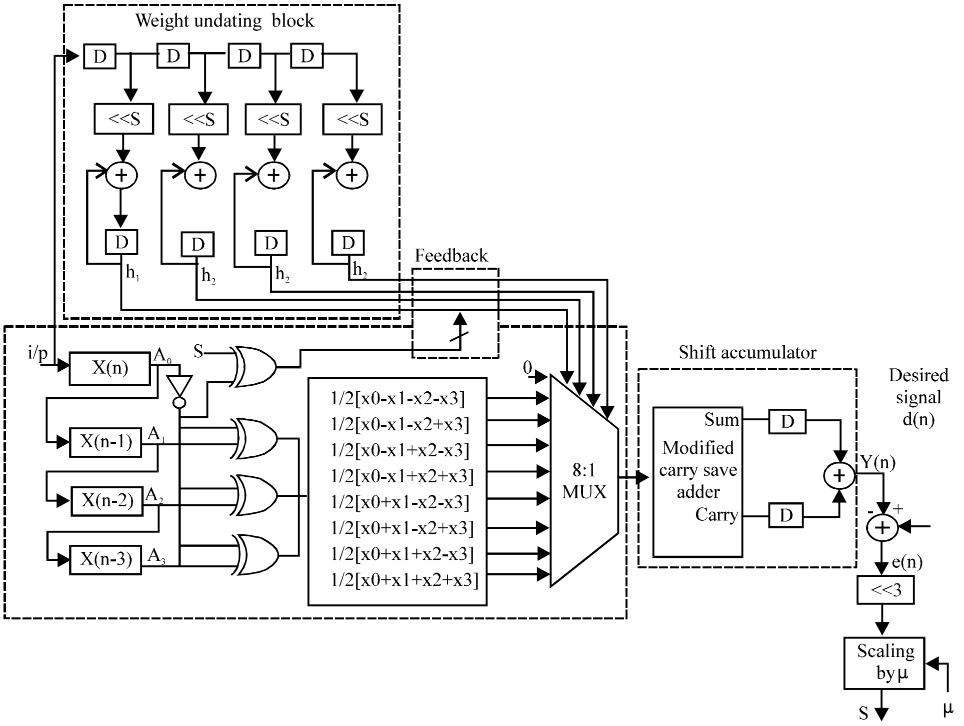

Fig. 4: Proposed OBC based DA in LMS adaptive FIR filter for N = 4

The famous adaptive filter called Least Mean Square (LMS) adaptive filter is used to find the filter coefficients of the Finite Impulse Response (FIR) filter. In the proposed method, the LMS adaptive filter has been acting as a weight updating block to calculate the filter coefficients of the FIR filter. The partial sum of input samples in the DA table and filter coefficients have been used as an address to access the DA table to reduce the computation complexity of LUT.

The inner product has 2^N possible values that are pre-computed and stored in 2^N words. For each sample, it is necessary to update LUT before filtering as well as the filter coefficients. The bit slices of filter coefficients obtained from the weight updating block have been fed to MUX from LSB to MSB. Hence, the output of the DA table fed to MCSLA where the number of area occupation has been reduced. The proposed structure of OBC based DA in LMS adaptive FIR filter of length N = 4 is shown in Fig. 4.

From Fig. 4, it consists of two blocks namely filter block and weight updating block. The filter block includes Distributed Arithmetic (DA) table performed using pipelined LUT and adders. 8:1 MUX is used to perform the MCSLA operation where the output of the DA table is set as a selection line for 8:1 MUX from the overall DA table outputs. After L clock cycles, the filter output y(n) has been obtained from MCSLA through the shifter and adder. Then the filter output y(n) has been subtracted from the desired signal d(n) to get the error signal e(n). The error signal e(n) has been scaled by using the convergence factor μ to produce the wired output as μe(n). The required output μe(n) has been again giving to the weight updating block as feedback. The output from the weight updating block has been given as a selection line to MUX. This process has been repeated continuously.

RESULTS AND DICUSSION

The architecture of the LMS adaptive FIR filter on the Offset Binary Coding (OBC) based scheme Distributed Arithmetic (DA) using Modified Carry Select Adder (MCSLA) has been computed using Quartus II 9.1 sp 1 Web Edition with the device as Stratix-EP2S15F484C3 and the implementation has been done by ModelSim XE III 6.3c. The simulation results of the conventional and proposed architecture for length N = 16 has been shown in Fig. 5 and 6. Similarly, the simulation results of the conventional and proposed architecture for length N = 32 have been shown in Fig. 7 and 8, respectively.

The comparison analysis for both conventional and proposed architecture is illustrated in Table 1. The synthesis results for conventional DA based LMS adaptive FIR filter architecture and the proposed OBC based DA scheme LMS adaptive FIR filter using MCSLA have been shown in Fig. 9.

Table 1: Comparison analysis of conventional and proposed architecture

Designs	Filter length (N)	DAT (nsec)	MUF (MHz)	No. of logic elements	No. of registers	Power (mW)	EPS (mW×nsec)
Conventional design	16	4.164	240.15	793	904	1.17	4.871
Park and Meher (2013)	32	4.694	213.04	2266	2152	1.23	5.773
Proposed design	16	3.564	280.58	442	483	1.14	4.062
	32	4.904	203.92	666	604	1.15	5.639

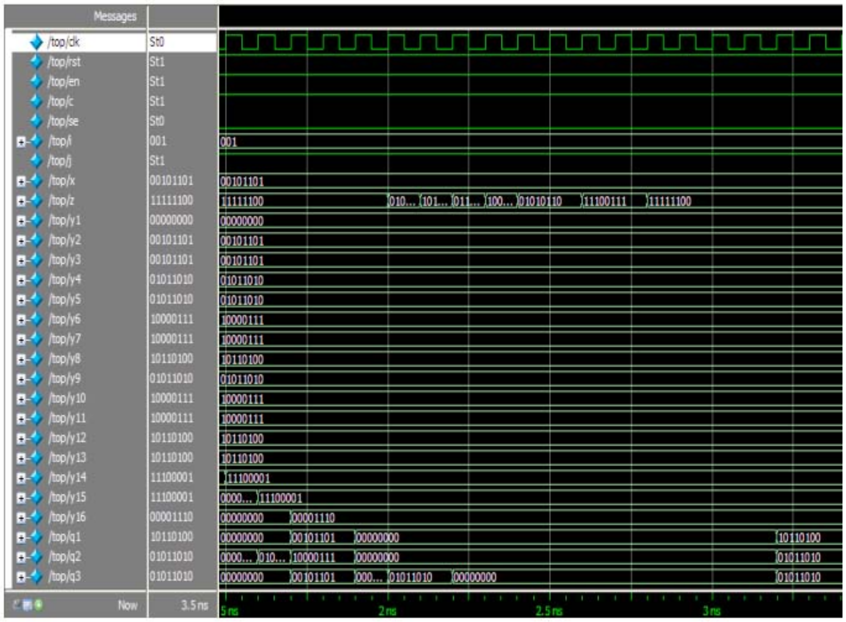

Fig. 5: Simulation result of conventional DA based LMS adaptive FIR filter for length N = 16

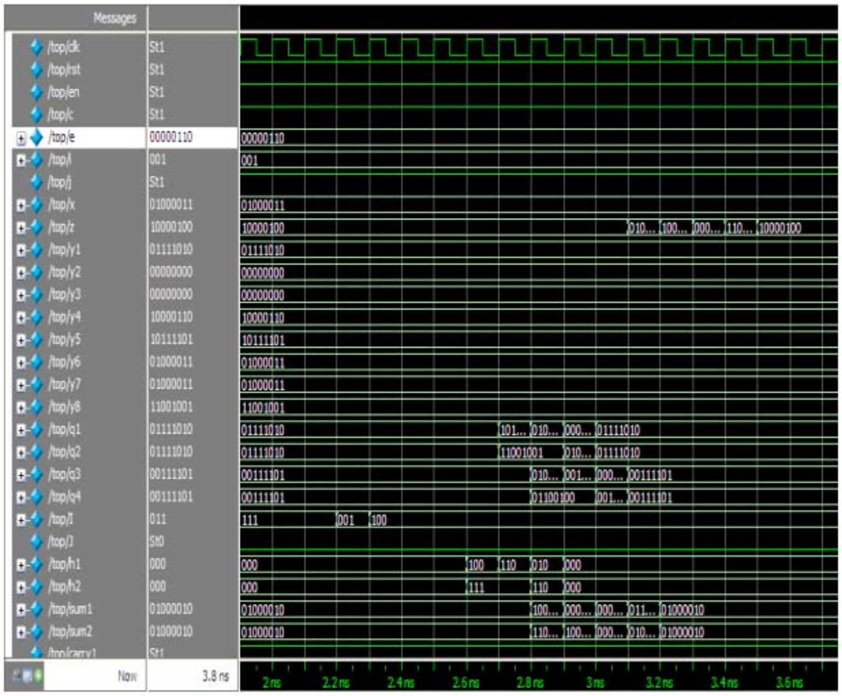

Fig. 6: Simulation result of proposed OBC based scheme DA architecture in LMS adaptive FIR filter for N = 16

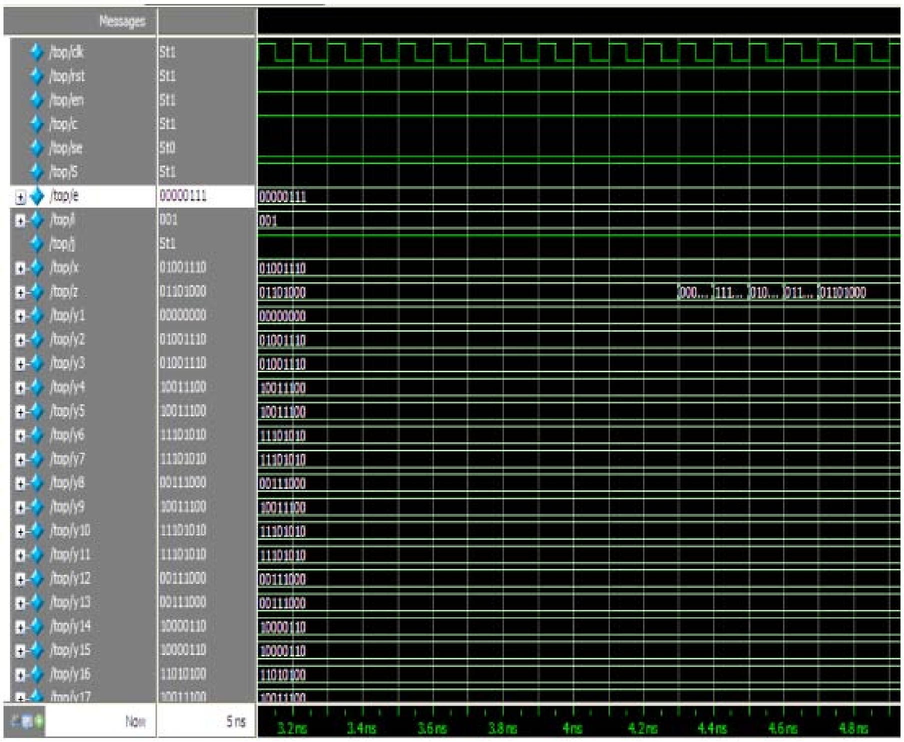

Fig. 7: Simulation result of conventional DA based architecture in LMS adaptive FIR filter for N = 32

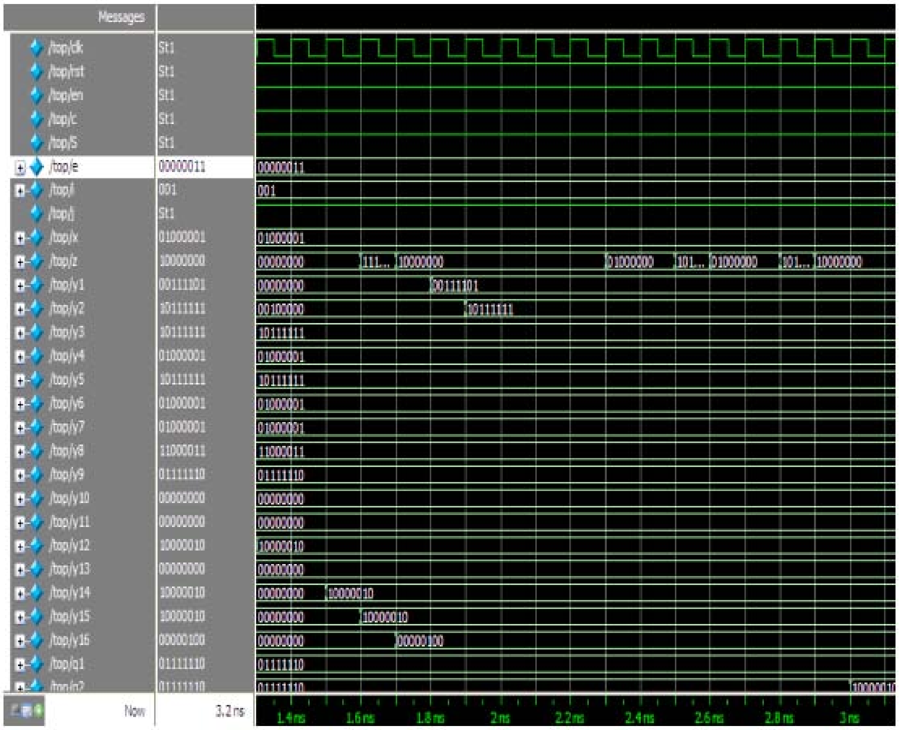

Fig. 8: Simulation result of proposed OBC based scheme DA architecture in LMS adaptive FIR filter for N = 32

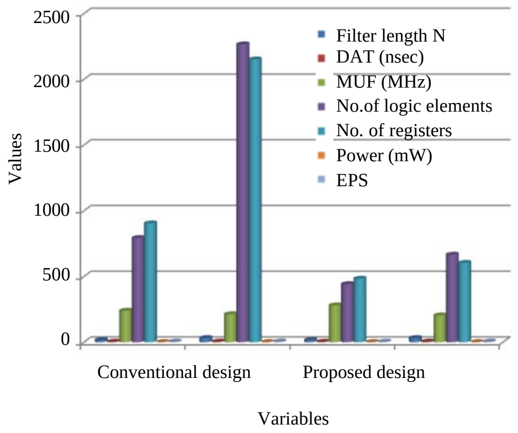

Fig. 9: Synthesis results of conventional and proposed architecture for N = 16 and 32

From Table 1 when compared to the conventional architecture, the proposed architecture for length N = 16 achieves 14.4% in DATs, 44.26% in No. of logic elements, 46.57% in No. of registers and 2.25% in power. Likewise, the proposed architecture for length N = 32 offers 4.28% in DATs, 70.6% in No. of logic elements, 71.93% in No. of registers and 6.5% reduction in power consumption.

CONCLUSION

The architecture of Offset Binary Coding (OBC) based Distributed Arithmetic (DA) in the Least Mean Square (LMS) adaptive FIR filter using Modified Carry Select Adder (MCSLA) for length N = 16 and 32 have been proposed using Quartus II 9.1 sp 1 Web Edition. The device used in Quartus II 9.1 sp 1 Web Edition is Stratix-EP2S15F484C3 and the simulation has been implemented in ModelSimXE III 6.3 c. When compared to the conventional DA based LMS adaptive FIR filter, the proposed OBC based DA in LMS adaptive FIR filter using MCSLA achieves better performance. OBC based DA in LMS adaptive filter achieves 44.26 and 70.6% reduction in No. of logic elements for length N = 16 and 32. In the future, it can be extended to 64 and 128 lengths to achieve good performance.

Significance statement: Form this method OBC based DA in LMS adaptive filter using MCSLA offer good performance interms of reducing area 44.26 and 70.6% in No. of logic elements for length N = 16 and 32. This method is used in the real time application for noise cancellation in telecommunication.

ACKNOWLEGEMENT

This research was supported by the University of Sathyabama Institute of Science and Technology under the guidance of Kalpalatha Reddy. We thank our colleagues from Panimalar Engineering College and Sathyabama University who provided insight and expertise that greatly assisted the research, although, they may not agree with all of the interpretations of this study.

REFERENCES

Allred, D.J., H. Yoo, V. Krishnan, W. Huang and D.V. Anderson, 2005. LMS adaptive filters using distributed arithmetic for high throughput. IEEE Trans. Circuits Syst. I. Regul. Pap., 52: 1327-1337.

Allred, D.J., W. Huang, V. Krishnan, H. Yoo and D.V. Anderson, 2004. An FPGA implementation for a high throughput adaptive filter using distributed arithmetic. Proceedings of the 12th International Annual IEEE Symposium on Field-Programmable Custom Computing Machines, April 20-23, 2004, IEEE, Napa, California, USA., pp: 324-325.

Chandekar, A.A. and M. Pawar, 2017. Delay and power optimized adaptive filter using distributed arithmetic. Proceedings of the 2017 International Conference on Electronics, Communication and Aerospace Technology (ICECA) Vol. 1, April 20-22, 2017, IEEE, Coimbatore, India, ISBN:978-1-5090-5685-9, pp: 256-261.

Guo, R. and L.S. De Brunner, 2011a. A novel adaptive filter implementation scheme using distributed arithmetic. Proceedings of the 2011 Conference Record of the 45th Asilomar Conference on Signals, Systems and Computers (ASILOMAR), November 6-9, 2011, IEEE, Pacific Grove, California, USA., ISBN:978-1-4673-0321-7, pp: 160-164.

Guo, R. and L.S. DeBrunner, 2011b. Two high-performance adaptive filter implementation schemes using distributed arithmetic. IEEE. Trans. Circuits Syst. II. Express Briefs, 58: 600-604.

Haykin, S. and B. Widrow, 2003. Least-Mean-Square Adaptive Filters. Vol. 31, Wiley, Hoboken, New Jersey, USA., ISBN:9780471215707.

Joanna, S.R. and A.V. Anathalakshmi, 2015. Design and implementation of efficient adaptive FIR filter based on distributed arithmetic. Proceedings of the 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), March 19-20, 2015, IEEE, Coimbatore, India, ISBN:978-1-4799-6817-6, pp: 1-4.

Mohanty, B.K., P.K. Meher and S.K. Patel, 2015. LUT optimization for distributed arithmetic-based block least mean square adaptive filter. IEEE. Trans. Large Scale Integr. Syst., 24: 1926-1935.

Park, S.Y. and P.K. Meher, 2013. Low-power, high-throughput and low-area adaptive FIR filter based on distributed arithmetic. IEEE. Trans. Circuits Syst. II. Express Briefs, 60: 346-350.

Park, S.Y. and P.K. Meher, 2014. Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter. IEEE. Trans. Circuits Syst. II. Express Briefs, 61: 511-515.

Prakash, M.S. and R.A. Shaik, 2013. Low-area and high-throughput architecture for an adaptive filter using distributed arithmetic. IEEE. Trans. Circuits Syst. II. Express Briefs, 60: 781-785.

White, S.A., 1989. Applications of distributed arithmetic to digital signal processing: A tutorial review. IEEE ASSP Magazine, 6: 4-19.