IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 7, Issue 4, Ver. II (Jul. - Aug. 2017), PP 31-37
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Low-Power, High-Throughput and Low-Area Adaptive Fir Filter Based On Distributed Arithmetic Using FPGA

*K.Indraja[1], M.Madhusudhan Reddy[2]
[1]*PG Scholar, Dept of ECE(VLSI&ES), GPREC(Autonomous), JNTUA, Kurnool, AP, India.*
[2] *Assistant Professor, Dept of ECE, GPREC (Autonomous), JNTUA, Kurnool, AP, India.*
Corresponding Author: K.Indraja

Abstract: *A unique pipelined architecture implementation of adaptive filter based on Distributed Arithmetic (DA) for low-power, high-throughput and low-space. Filtering process involves more space and is not used for higher order filters, therefore it provides a great attrition in the throughput. These are problems that have been overcome by effectual DA formulation of adaptive filters. Distributed Arithmetic is an efficacious procedure for calculating the inner products between a fixed and a variable data vector. Equivalent appliance of 4-point inner product and coefficients increments units to produce immense throughput rate. Carry-save accumulation is used in order to rebate the sampling period and expanse complication for DA-based inner-product estimation. Power drain is reduced by using bit clock for carry-save accumulation when compared to all other process. To amend the weights by using least mean square (LMS) adaption and also attenuate the mean square error between the calculated and desired output. It rebate the LUT's, available flip-flop slices for this approach.*

Keywords: *Adaptive FIR filter, Carry save accumulation, DA, Least Mean Square theorem.*

Date of Submission: 14-08-2017 Date of acceptance: 05-09-2017

I. Introduction

The proliferation of Adaptive filters find wide appliances in several digital signal processing (DSP) areas, like noise and echo cancellation, system identification, channel equalization and calculation. Generally, filter plays a major sole for removal of unwanted signal/ noise from the original signal, specifically, Adaptive FIR filter is simple to attract for many applications, where it is needed to reduce process requirement.

The operation of adaptive filter is based on the calculation of the statistical properties of the signal in the environment while modifying the value of its parameters in order to rebate a such criterion function. the statistical parameters vary with time. It consists of two process i.e., filtering process and the coefficients updating process. The Adaptive filter theorems are Least Mean Square (LMS), Recursive Least Square (RLS), Normalized Least Mean Square (NLMS). The LMS algorithm is a class of adaptive filter to mimic a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signal. The RLS algorithm includes many complicated mathematical procedure and NLMS algorithm is used in system in order to elevate the voice quality. Finally, LMS algorithm is most suitable for reducing the power and space. The mien features that attracted the use of the LMS algorithms are computational complexity and also it imply a feedback connection. The mien asset of LMS algorithm is its simplicity in terms of memory requirement. DA is basically a bit serial computation process that forms an inner product of pair of vectors in a single direct step. DA efficiently implements the MAC using basic building blocks in FPGAs. By using DA instead of traditional way, a huge abatement in area can be achieved.

DA-based scheme of adaptive filter has been suggested by Surya Prakash and Rafi Ahamed Shaik [10], have evaluated the design by using two LUTs to store weights and input samples during each iteration and also Mohanty and Meher [13] proposed a BLMS algorithm, in which this design makes use of a LUT sharing technique to get filter output.

II. Adaptive Lms Algorithm

Adaptive filters are dynamic filters which change their characteristics to achieve desired yield. Adaptive filter also has a various parameters which need to be changed in order to maintain optimal filtering. An adaptive filter may be understood as a self-modifying digital filter that adjust its coefficients in order to minimize an error function. This error function, also referred to as cost function, is a distance measurement between the reference or desired signal and output of the adaptive filter. LMS algorithms have been applied to an extensive number of problems including signal prediction, adaptive arrays e.tc., The basic configuration of an adaptive filter is shown in fig.1 shows which consists of general Adaptive FIR filter block and weight update

block. In such a scheme, the input signal is denoted by x(k), the reference signal d(k) represents the desired output signal, y(k) is the output of the adaptive filter.

The error signal e(k) is used by the adaption algorithm to update filter coefficients vector w(k) according to equation (1a),

$$w(k+1)=w(k)+\mu .e(k).x(k) \quad (1a)$$

where,

$$e(k)=d(k)-y(k) \quad (1b)$$

$$y(k)=w^{qt}(k).x(k) \quad (1c)$$

x(k) and w(k) of an N^{th} order LMS adaptive filter at the nth iteration respectively and is given by,

$$x(k)=[x(k), x(k-1)..... x(k-N+1)]^T \quad (2a)$$

$$w(k)=[w_0(k), w_1(k).....,w_{K-1}(N)]^T \quad (2b)$$

e(k) is the error computed during the k^{th} iteration, which is used to update the weights, μ is the convergence factor and N is the length of filter.

In the DLMS algorithm, instead of using the recent-most feedback error e(k) corresponding to the n^{th} iteration for updating the filter weights, it uses the delayed error e(k-m), that is the error corresponding to $(k-m)^{th}$ iteration for updating the current weight. The coefficient-update equation of such delayed LMS adaptive filter is given by,

$$w(k+1)=w(k)+\mu.e(k-m).x(k-m). \quad (3)$$

where m is the "adaption delay".

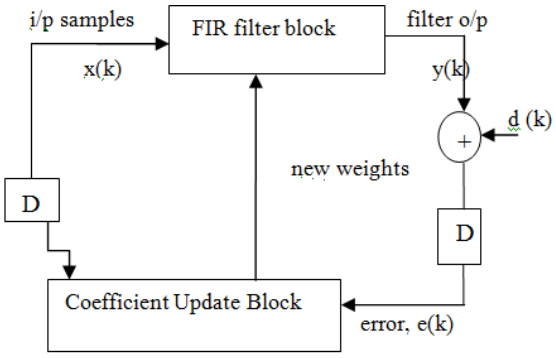

Fig.1.General Adaptive Filter

III. Distributed Arithmetic

Distributed Arithmetic is generally used for signal processing theorems where computing the inner product of two vectors comprises most of the computational work load. Ironically, many DSP designs have never heard of this algorithm. Inspired by the potential of the Xilinx FPGA look up table architecture, the DA was resurrected in the early 90's and shown to produce very effectual filter design. It is one of the efficacious approach for realization of high order filters as it can achieve the large throughput without the help of hardware multiplier.

The LMS adaptive filter, is necessary to perform an inner product computation in each cycle, the most of the critical path is contributed by this. DA inner product generation considers the estimation of the following sum of products,

$$y = \sum_{k=0}^{N-1} w_k \cdot x_k \quad (4)$$

where w_k denotes the coefficients of the filter, x_k denotes the inputs of the filter. If we consider bit width of the coefficient is L, each component in the weight vector may be represented by two's complement form, such that $|w_k|<1$, then we may express w_k as,

$$w_k = -w_{k0} + \sum_{l=1}^{L-1} w_{kl} \cdot 2^{-l} \quad (5)$$

Now by combining the equations (4) and (5) the expression y in terms of bits of w_k are shown below: Equation (6) is the conventional form of expressing the inner product computation. By interchanging the summation order, we get,

$$y= -\sum_{k=0}^{N-1} x_k \cdot w_{k0} + \sum_{k=0}^{N-1} x_k \cdot \sum_{l=1}^{L-1} w_{kl} \cdot 2^{-l} \quad (6)$$

The following equation defines a distributed arithmetic computation.

$$y = [\sum_{l=1}^{L-1} 2^{-l} \cdot y_l] - y_0 \quad (7)$$

Because each w_k may be take on values of 0 and 1 only, equation (5) may have only 2k possible values. Rather than compute these the values and store in a ROM. The input data can be used directly to address the memory. After N such cycles, the memory contains the result.

IV. Proposed Schemes

In the proposed approach, for the implementation of the DA based adaptive filter, we make use of the DA-table structure that makes use of the 8 delay elements. Hence by using the proposed configuration, we can reduce the original DA-table structure that increases the area efficiency of the design. The proposed structure of D-table which makes use of 8 delay elements is shown in Fig.4. Therefore, we are designing a advanced DA based architecture for 4bit and 16bit LMS adaptive filters.

4.1.Proposed structure of DA Based Adaptive Filter of filter length N=4

The DA based adaptive filter of length N=4 consists of inner product block of four points, additional circuitry for coefficient increment block, computation of error value e(k), and the circuitry to generate the control word 't' which is further used by the barrel shifter are shown in Fig.2. The 4-point inner product block which was shown in Fig.3, consists of a DA-table which is used in this consists of a 15 register array. The existing DA-based LMS adaptive filter design requires 16 delay element DA table structure. But the same structure can be implemented with the help of 8 delay elements.

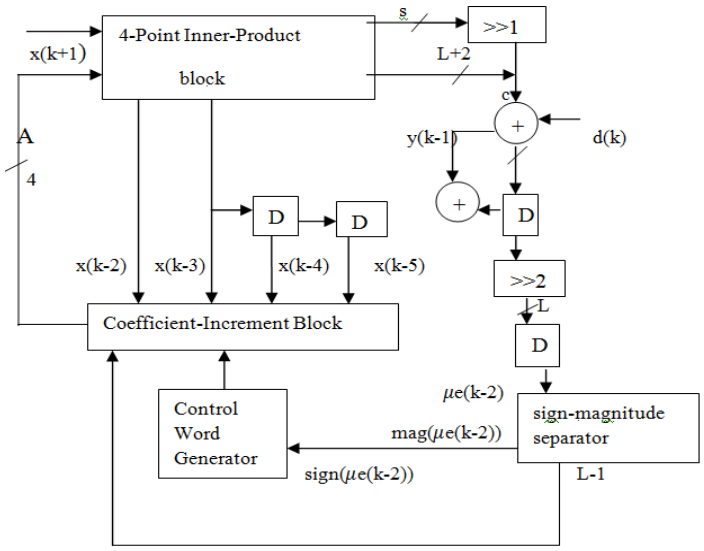

Fig.2. Proposed structure of DA based adaptive filter of filter length N=4.

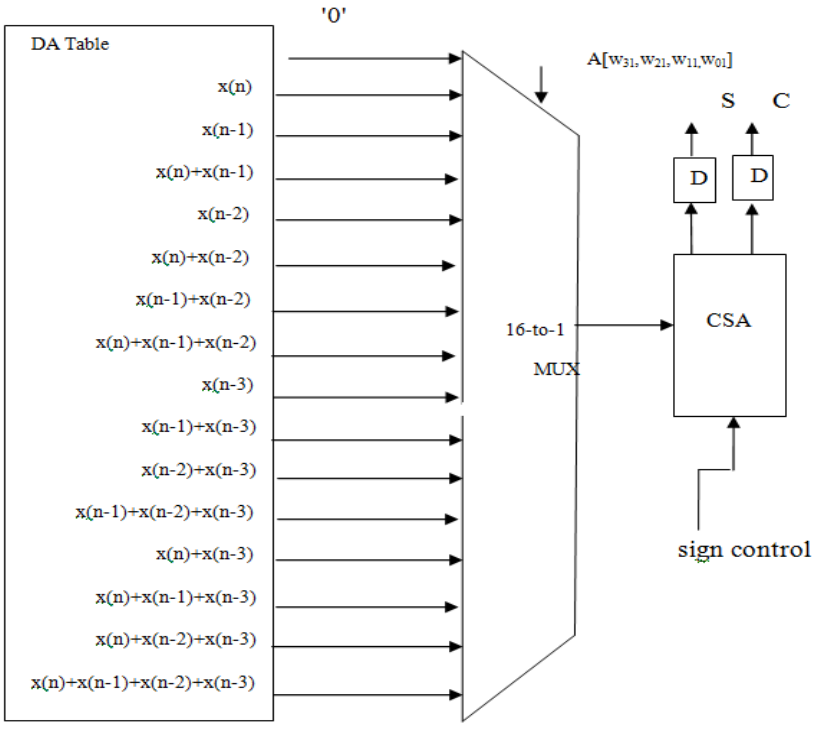

Fig.3. Structure of 4-Point Inner Product block

DA table is used to reduce the number of clock cycles required to compute the the sum of input words. It is capable of storing the partial inner products 'y_l' for $0<l\leq15$. In order to store the pre-computed sums of partial products of input words, it contains only 15 registers. And a 16:1 multiplexer is used to select the contents present in the register and also the weights or Coefficients A={w_{3l},w_{2l},w_{1l},w_{0l}} are fed to the multiplexer as control bits in the LSB-MSB order. The yield of the MUX is then connected to the CSA block, after every L bit cycles. The inner products can be calculated in L cycles by shift and accumulation, followed by look up table operations corresponding to L number of bit slices {w_{kl}} and all the partial inner products which are fed from the MUX and it fructifies a sum and a carry words. The sum words are shifted and then added with the carry words and an input carry '1' to generate the filter output. This filter yield is subsequently subtracted from the wanted signal d(k) to obtain the error e(k). After that the sign-magnitude separator is used to separate the sign bit and magnitude bits from the obtained error.

The magnitude bits are used by the control word generator to shifter. The logic used for the procreation of control word 't' for barrel shifter is shown in Fig.5. The convergence factor 'μ' is taken as 1/N. Generally, we can take μ as 2^{-i}/N, where 'i' is a small integer. The coefficient increment unit for N=4 consists of four barrel shifters and four adder/subtractor cells. The configuration of coefficient increment block is shown in Fig.6.

In general, Barrel Shifter is the most helpful unit in digital signal processing appliances. It performs multiplications just by right shifting the data in registers by required number of times. That means to multiply the data by 2 times, the data is shifted right by one position. Similarly, division can be performed by left shifting the data as per claim. Thus the barrel shifter reduces the complexity of operations there by area optimization has been achieved too. Here the barrel shifter shifts the input values $\boldsymbol{x_k}$, where k=0,1,----N-1 by defined number of locations according to estimated error value. The barrel shifter produces the desired values that need to be added or subtracted from the content of the weight corresponding current value in register. The word parallel bit serial converter is used to convert the sum coming from the 4 adder and subtractor blocks load parallely and convert it into serial format which are used as weights to select the pre partial products from the DA table.

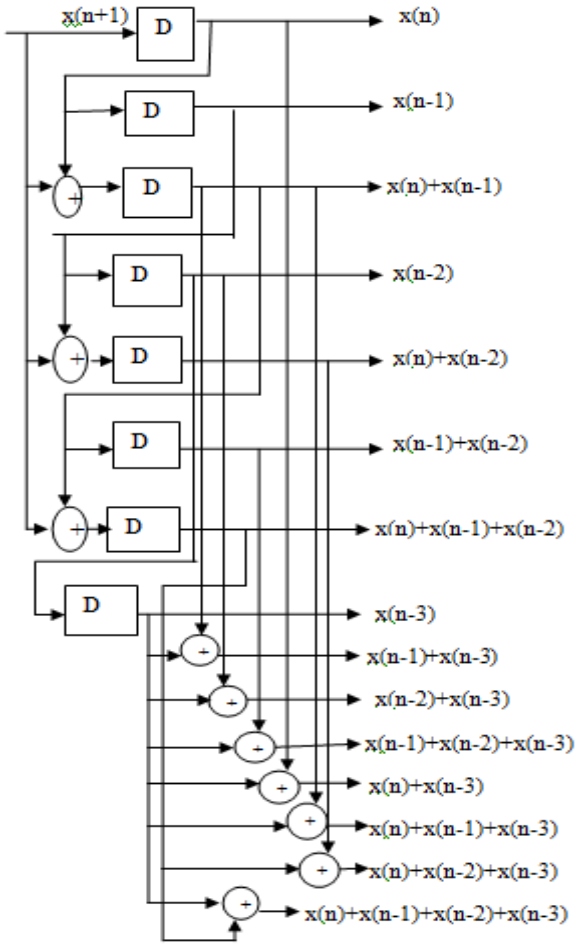

Fig.4. DA table for generation of possible sum of input samples block

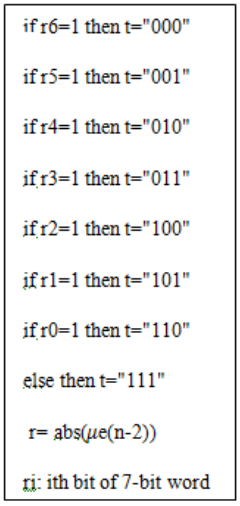

Fig.5. Logic used for the control word t for BS

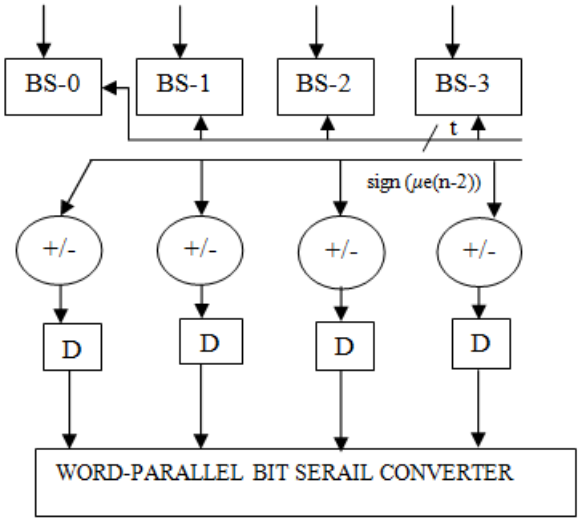

Fig.6.Structure of coefficient increment

4.2.Proposed structure of DA based Adaptive filter of filter length N=16

The architecture of DA based adaptive filter of N=16 is shown in Fig.7. This design includes four sets of 4-point inner product blocks and coefficients increment blocks. All the blocks are connected in proper manner in order to give wanted result. The four 4-point inner product blocks and coefficients increment blocks all together is known as 16-bit data computing block as this sub block computes 16-bit sum and carry words. These are used in further computations. As in the case of fourth order filter appliances, here also the sums and carry are produced by the four inner product blocks. And these will be added by using two binary adder trees. The output of four 4-point inner product blocks i.e., sum words are added with 4 carry in bits. Since the carry words are of double the coefficient compared to the sum words, 2 carry in bits are used as input carry at 1stlevel and carry words of binary adder tree, this contains a 4 carry in bits to sum words. Sign magnitude separator is used to divide the sign and magnitude are taken from the error which is calculated in case of filter N=4. The outputs that are obtained from the sign-magnitude separator and 't', which is used as a control word generator for barrel shifter are generally connected to all the coefficient increment blocks.

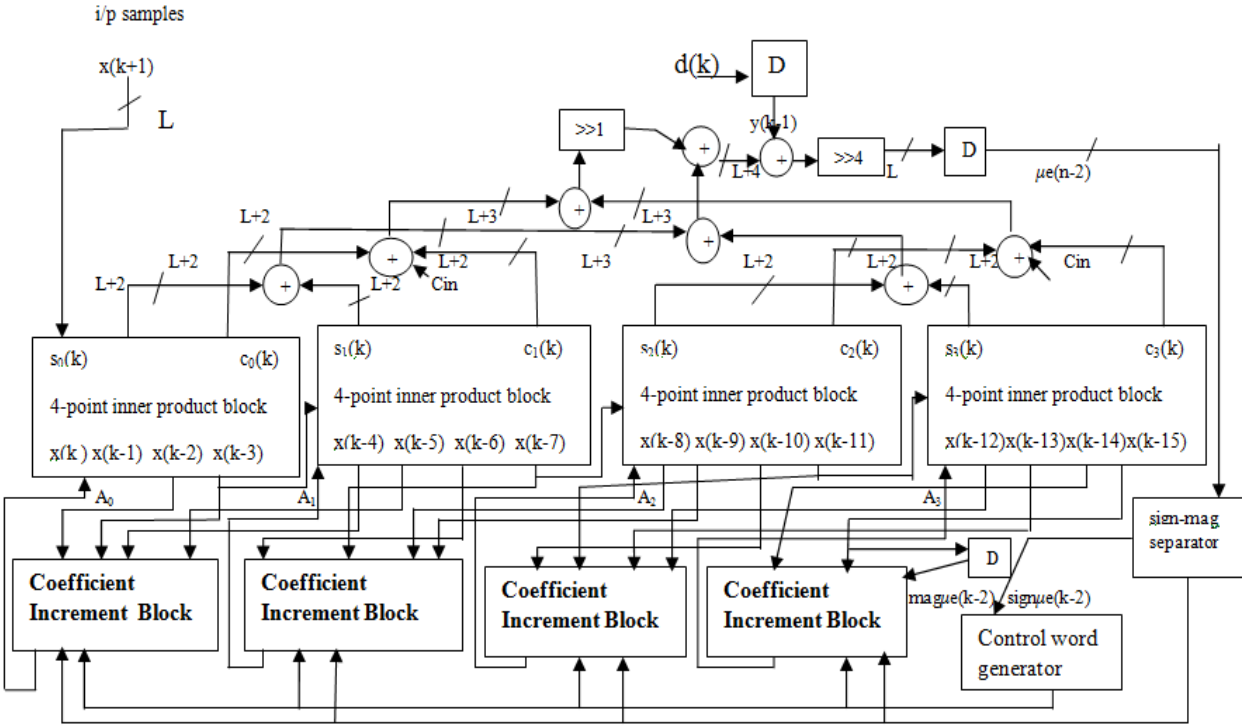

Fig.7.Proposed structure of DA based Adaptive filter of filter length N=16

V. Synthesis &Simulation Results

The proposed design has been simulated using Xilinx 14.5 and ModelSim 6.8.b, the waveforms obtained after simulating is as shown in Fig.9,11, synthesis reports for 4-tap, 16-tap filters are shown in Fig.8,10., and also the proposed design simulated result is implemented on FPGA Spartan 3E board, is a high volume Starter board gives designers instant access to complete platform capabilities of Spartan 3E family.

Project File:	as.xise	**Parser Errors:**	No Errors
Module Name:	top_module_adaptive_filter	**Implementation State:**	Programming File Generated
Target Device:	xc3s500e-4ft256	**• Errors:**	No Errors
Product Version:	ISE 14.5	**• Warnings:**	107 Warnings (107 new)
Design Goal:	Balanced	**• Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	All Constraints Met
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	271	9,312	2%	
Number of 4 input LUTs	471	9,312	5%	
Number of occupied Slices	296	4,656	6%	
Number of Slices containing only related logic	296	296	100%	
Number of Slices containing unrelated logic	0	296	0%	
Total Number of 4 input LUTs	477	9,312	5%	
Number used as logic	471			
Number used as a route-thru	6			
Number of bonded IOBs	30	190	15%	
Number of BUFGMUXs	3	24	12%	
Average Fanout of Non-Clock Nets	3.55			

Fig.8.Synthesis report for 4-tap adaptive filter

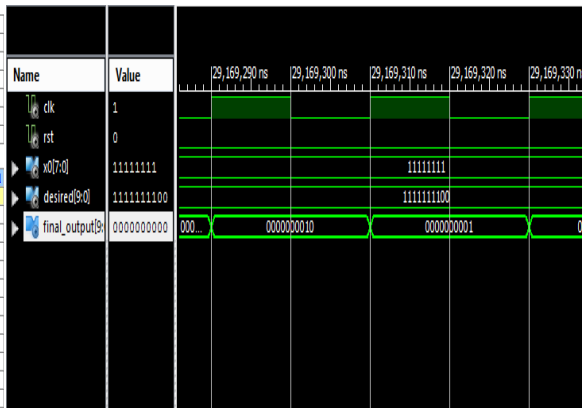

Fig.9. Simulation output for 4-tap adaptive filter

Project File:	fv.xise	**Parser Errors:**	No Errors
Module Name:	top_adap16tap	**Implementation State:**	Placed and Routed
Target Device:	xc3s500e-4ft256	**• Errors:**	No Errors
Product Version:	ISE 14.5	**• Warnings:**	394 Warnings (394 new)
Design Goal:	Balanced	**• Routing Results:**	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	**• Timing Constraints:**	All Constraints Met
Environment:	System Settings	**• Final Timing Score:**	0 (Timing Report)

Device Utilization Summary

Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	912	9,312	9%	
Number of 4 input LUTs	1,523	9,312	16%	
Number of occupied Slices	959	4,656	20%	
Number of Slices containing only related logic	959	959	100%	
Number of Slices containing unrelated logic	0	959	0%	
Total Number of 4 input LUTs	1,544	9,312	16%	
Number used as logic	1,523			
Number used as a route-thru	21			
Number of bonded IOBs	34	190	17%	
Number of BUFGMUXs	6	24	25%	
Average Fanout of Non-Clock Nets	3.15			

Fig.10.Synthesis report for16-tap adaptive filter

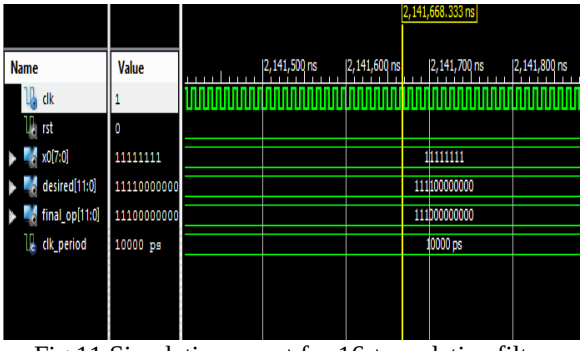

Fig.11.Simulation ouput for 16-tap adative filter

VI. Conclusion

In this paper, an adaptive FIR filter based on distributed arithmetic using FPGA for area efficacious method is implemented and also the reduction in power consumption is achieved too. High throughput is intensely enhanced by parallel LUT update and filtering and coefficient update operations performing synchronously. The carry save accumulation operation is used to produce the signed partial inner products, which compute the filter output very fast and also offset binary coding is popularly used to reduce the LUT size to half for area-efficient implementation of DA.

Acknowledgment

The authors would like to express their gratitude to **Pramod Kumar Meher**, Ph.D.degree in science from Sambalpur University, India and also **Sang Yoon Park**, Ph.D.degree in electrical engineering and computer science from Seoul National University, Seoul, Korea, for providing their valuable brief i.e., "**Low-power, High-throughput and Low-area adaptive Fir filter based on DA"**, during this project work.

References

[1] S.Y.Park & P.K.Meher," Low-power, High-throughput and Low-area adaptive Fir filter based on DA", IEEE Trans. on Ckts. and Syst. 2015.
[2] S.Haykin, "Least Mean Square Adaptive filters", Hoboken, NJ, USA: Wiley, 2003.
[3] A.Peled and B.Liu, "A new hardware realization of digital filters", IEEE Transactions on Acoustics, Speech and signal processing, vol.22, pp.456-462, December 1974.
[4] S.A.White ,"Applications of the distributed arithmetic to digital signal processing: A tutorial review," IEEE ASSP Mag., vol6,no.3,pp.4-19,July 1989.
[5] Q.Zhu, S.C.Douglas and K.F.Smith, "A pipelined architecture for LMS Adaptive FIR filters without adaption delay", 0-8186-7919-0/97 1997 IEEE.
[6] D.J.Allred, H.Yoo, V.Krishna, W.Huang and D.V.Anderson,"LMS adaptive filters using DA for high throughput", IEEETrans.Circuits,vol.52,no.7.pp.1327-1337, July 2005.
[7] B.K.Mohanty and P.K.Meher,"Delayed block LMS algorithm and concurrent architecture for high-speed implementation of Adaptive FIR filters", IEEE Trans., November 2008.
[8] P.K.Meher and M.Maheswari,"A high speed FIR adaptive filter architecture using a modified delayed LMS algorithm", in Proc. IEEE International Symposium on Circuits and Systems (ISCAS'2011), Rio de Janerio, Brazil, May 2011, pp121-124.
[9] R.Guo & L.S.DeBrunner, "Two high performance adaptive filter implementation schemes using DA", IEEE Trans.Circuits Syst.II, Exp. Briefs,vol.58,no.9, pp.600-604,Sptember 2011.
[10] R.Guo & L.S.Debrunner, "A novel adaptive filter implentation schemes using distributed arithmetic", Proc. Asilomar Conf.Signals, Syst., comput.,Nov.2011,pp.160-164.
[11] M.Surya Prakash and Rfi Ahamed Shaik,"Low-area and High- Throughput architecture for an adaptive filter using DA", IEEE Trans. on ckts.and Syst. regular papers, vol.61,no.3, November 2011.
[12] Anirut Trakultritrung, Ekkawin Thanangchusin, Sorawat Chivapreecha, "Distributed Arithmetic LMS Adaptive Filter Implementation without Look-Up Table", 978-1-4673-2025-2/12/ 2012 IEEE.
[13] B.K.Mohanty and P.K.Meher,"A High performance energy efficient architecture for FIR adaptive filter based on DA of BLMS algorithm", IEEE Trans. on signal processing, vol.61, no.4, Feb 2013.
[14] P.K.Meher and S.Y.Park, "Critical-Path Analysis and Low-Complexity Implementation of the LMS Adaptive Algorithm", IEEE Transactions On Circuits And Systems regular papers, vol.61,no.3,March,2014.

IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) is UGC approved Journal with Sl. No. 5081, Journal no. 49363.

K.Indraja. "Low-Power, High-Throughput and Low-Area Adaptive Fir Filter Based On Distributed Arithmetic Using FPGA ." IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) , vol. 7, no. 4, 2017, pp. 31–37.