ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 9, September 2015**

# Adaptive Fir Filter with Optimised Area and Power using Modified Inner-Product Block

Jesmin Joy

M. Tech Scholar (VLSI & Embedded Systems), Dept. of ECE, IIET, M. G. University, Kottayam, Kerala, India

**ABSTRACT**: An adaptive filter is a digital filter that can adjust its coefficients to give the best match to a given desired signal. When an adaptive filter operates in a changeable environment the filter coefficients can adapt in response to changes in the applied input signals. The choice of the filter structure and the criterion function used during the adaptation process, have the crucial influence to the characteristics of the adaptive Filter as a whole. A new design and implementation of FIR filters using Distributed Arithmetic is provided in this paper. Distributed Arithmetic structure is used to increase the resource usage while pipeline structure is also used to increase the system speed. The memory size can be reduced by decomposing the LUT. FIR filter is designed using multiplexer which is used to select the filter coefficients. The techniques used in this paper provide reduction in LUT size compared with the conventional Look up Table (LUT) of adaptive FIR filter. By the proposed method, area efficiency, low power and high throughput is achieved.

**KEYWORDS**: adaptive filter; least mean square (LMS); distributed arithmetic (DA); finite impulse response

## I. INTRODUCTION

LMS based adaptive filters are preferred for most of the DSP applications. The goal of the adaptation is to adjust the characteristics of the filter through an interaction with the environment in order to reach the desired values. The operation of adaptive filters is based on the estimation of the statistical properties of the signal in its environment, while modifying the value of its parameters in order to minimize a certain criterion function. The criterion function may be determined in a number of ways, depending on the particular purpose of the adaptive filter, but usually it is a function of some reference signal. The reference signal may be defined as the desired response of the adaptive filter, and in that case the role of the adaptive algorithm is to adjust the parameters of the adaptive filter in such a way to minimize the error signal, which represents the difference between the signal at the output of the adaptive filter and the reference signal. When we are doing the direct form configuration of the filters it leads to long critical path because of the inner-product computation to get the filter output. Hence for high sampling rate, the critical path of structure should not exceed the sampling period. So, we go for distributive arithmetic (DA).

DA is basically a bit serial computational operation that forms an inner (dot) product of a pair of vectors in a single direct step. The advantage of DA is its efficiency of mechanization. In the multiplier-less distributed arithmetic (DA)-based technique has gained substantial popularity for its high-throughput processing, which result in less cost and area efficient computing structure. This brief proposes a novel DA-based architecture for low-power, low-area and high-throughput pipelined implementation of adaptive filter with very low adaptation delay. Conventional adder-based shift accumulation is replaced by a conditional carry-save accumulation of signed partial linear products to reduce the sampling period.Finite-impulse-response (FIR) filters are basic processing elements in applications such as video signal processing and audio signal processing. Adaptive digital filters have been applied to a wide variety of important problems in recent years. Perhaps one of the most well known adaptive algorithms is the least mean squares (LMS) algorithm, which updates the weights of a transversal filter using an approximate technique of steepest descentMany applications in digital communication (channel equalization, frequency channelization), speech processing (adaptive noise cancelation), seismic signal processing (noise elimination), and several other areas of signal processing require large order FIR filters. Since the number of multiply-accumulate (MAC) operations required per filter output increases linearly with the filter order, real-time implementation of these filters of large orders is a challenging task.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 9, September 2015**

## II. LITERATURE SURVEY

Adaptive digital filters have tremendous applications in signal processing. According to the LMS algorithm, there is a delay in the feedback error for updating the weights which does not favour the pipeline implementation, when the sampling rate is high. [2] Have proposed the delayed LMS algorithm for pipeline application of LMS based ADF. In LMS algorithm, the adaptation step can be performed after a fixed delay only, in some practical situations. In such a cases the implemented algorithm is a modified version of LMS algorithm known as the delayed LMS (DLMS) algorithm. In DLMS the coefficient adaptation is performed after a delay. The work shows the conditions for convergence and estimates of convergence rate, both for the mean of the DLMS filter coefficients and for its excess mean square error. The only difference between the LMS and DLMS algorithm is that the correction term for updating the filter weights of the current iteration are computed from error corresponding to the past iteration. Many methods have been proposed to implement BLMS based adaptive digital filters efficiently in systolic VLSI with minimum adaptation delay [2].

In order to avoid adaptation delay [3] has proposed a modified DLMS algorithm. In some of the applications of the adaptive finite impulse response filtering, the adaptation algorithm can be applied with a delay only in the coefficient update. This has a dissimilar effect on the convergence behaviour of the algorithm. In this work it is shown how the delayed LMS algorithm can be converted into the typical LMS algorithm at simply slight increase in the computational expense.

The modified DLMS is used by [4] to derive a systolic architecture but it requires large amount of hardware resources as compared to the earlier one. BLMS is useful for fast and computationally-efficient implementation of adaptive digital filters. The convergence performance of BLMS ADFs and LMS ADFs are similar, but for block length $L$ BLMS ADFs offers $L$ fold higher throughput. Considering this, many BLMS algorithms like time and frequency domain block filtered-X LMS (BFXLMS) has been proposed for specific applications. Computationally more efficient BFXLMS using FFT and fast Hartley transform (FHT). A delayed block LMS algorithm and a concurrent multiplier-based design for high throughput pipeline execution of BLMS ADFs have been proposed.

In a block LMS algorithm with delayed weight adaptation for hardware execution of FIR adaptive filters has been proposed. The delayed block least mean square algorithm take a block of $L$ input samples and produces block of $L$ output, in every training cycle. The simulation result shows that the DBLMS algorithm has convergence performance same as that of the DLMS algorithm. A highly synchronized systolic architecture for FIR adaptive filters has been derived. The suggested architecture can support $L$ time higher sampling rate when compared with the other pipelined designs and hence include less samples of adaptation delays and would provide a more effective execution of LMS-based adaptive filters. [9], [10] have suggested structure for FPGA implementation of BLMS ADFs based on distributed arithmetic. [9] Derived a design and implement a high throughput ADF using Fast Block Least Mean Squares (FBLMS) adaptive algorithm. The structure of filter is built on Distributed Arithmetic. The structure calculate the inner product as: (i) shifting (ii) accumulating (iii) storing in look-up table. The desired adaptive digital filter obtained will be multiplier less. Hence a DA based execution of adaptive filter is area efficient. FPGA implementation results imitates that the proposed DA based adaptive filter in [9] can implement with meaningfully smaller area usage, (about 45%) less than that of the remaining FBLMS algorithm based adaptive filter. The structure in [10] replaces multiply-and accumulates operations with a series of look-up-tables (LUT). FPGA implementation results in [10] conforms that the suggested DA based adaptive filter can implement with expressively smaller area usage about 52% less than that of the existing FBLMS algorithm based adaptive filter applications. The structure in [8] for block LMS ADFs supports a very low sampling rate because it uses single multiply-accumulate cell for the computation of filter output and the weight increment term.

DA uses bit-serial operations and LUTs to implement high throughput filters which uses simply about one cycle per bit of resolution irrespective of filter length. Though, building adaptive DA filters requires recalculation of the LUTs for every adaptation which can deny any performance advantages of DA filtering. With the help of an auxiliary LUT with distinctive addressing, the efficiency and throughput of DA adaptive filters can be made as same order as fixed DA filters. In this paper, a new hardware adaptive filter structure has been suggested for very high throughput LMS adaptive filters have described the development of DA adaptive filters and showed that practical executions of DA

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 9, September 2015**

adaptive filters have very high throughput comparative to multiply and accumulate architectures also showed that DA adaptive filters have a potential area. The power consumption advantage over digital signal processing microprocessor architectures is also achieved.

## III. PROPOSED ADAPTIVE FILTER

The basic block diagram for the proposed system for the design of an Adaptive FIR Filter using Distributive Arithmetic is as shown below:

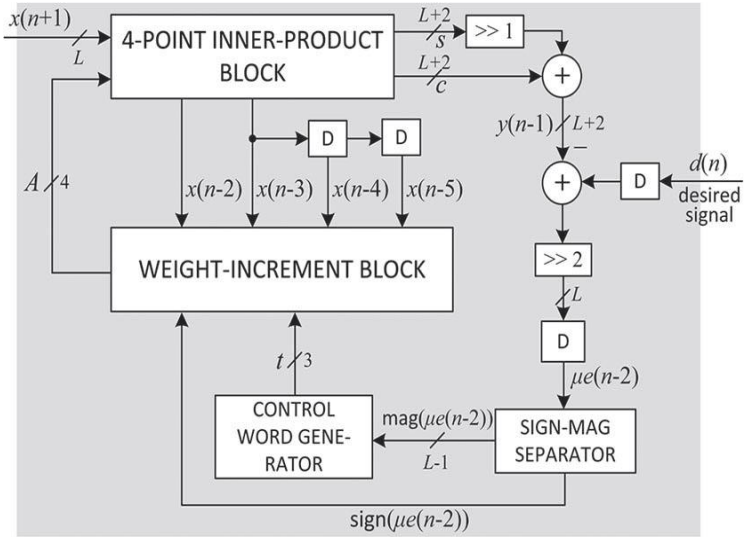


Fig. 1. Proposed DA-Based LMS Adaptive Filter

The proposed structure of DA-based adaptive filter of length $N = 4$ is shown in Fig.1. It consists of a four-point inner-product block and a weight-increment block along with additional circuits for the computation of error value $e(n)$ and control word $t$ for the barrel shifters.On comparing with existing DA based adaptive filter implementations the proposed architecture has achieved low complexity .For achieving this advantage Offset Binary Coding (OBC) has been used in this filter design. By doing so, the number of LUTs used has been decreased from 16 to 8.The existing DA-based LMS adaptive filter design requires 16 delay element DA-table structure. In the proposed structure there is significant reduction in the number of delay elements and adders.

### A. *Four Point Inner-Product Block:*

The initial part of the filtering process of the LMS adaptive filter, for each cycle, is the need to perform an inner-product computation. Distributed Arithmetic (DA) technique is bit-serial in nature. It is actually a bit-level rearrangement of the multiply and accumulation operation. The basic DA is a computational algorithm that affords efficient implementation of the weighted sum of products, or dot product. DA is a bit-serial operation used to compute the inner (dot) product of a constant coefficient vector and a variable input vector in a single direct step .This is the task that contributes to most of the critical path .Let the inner product be assumed to be:

$$y(n)=W^{qT}(n).X(n)$$
$$e(n)=d(n)-y(n)$$

where input vector x($n$) and the weight vector w($n$) at the $n$th iteration are respectively given by

$$X(n)=[x(n),x(n-1),\ldots\ldots x(n-N+1)]^{T}$$
$$W(n)=[w_0(n), w_1(n),\ldots., w_{N-1}(n)]^{T}$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 9, September 2015**

$d(n)$ is the desired response and $y(n)$ is the filter output of the $n$th iteration ,$e(n)$ denotes the error computed during the $n$th iteration, which is used to update the weights. The weights of LMS adaptive filter during the $n$th iteration are updated according to the following equation

$$W(n+1)=w(n)+\mu.e(n-m).X(n-m).$$

$\mu$ is the convergence factor, and $N$ is the filter length. In this case pipelined designs, the feedback error $e(n)$ is available after certain number of cycles called as the "adaptation delay." The pipelined architectures used for the delay error $e(n - m)$ for updating the current weight instead of the most recent error, where $m$ is the adaptation delay.

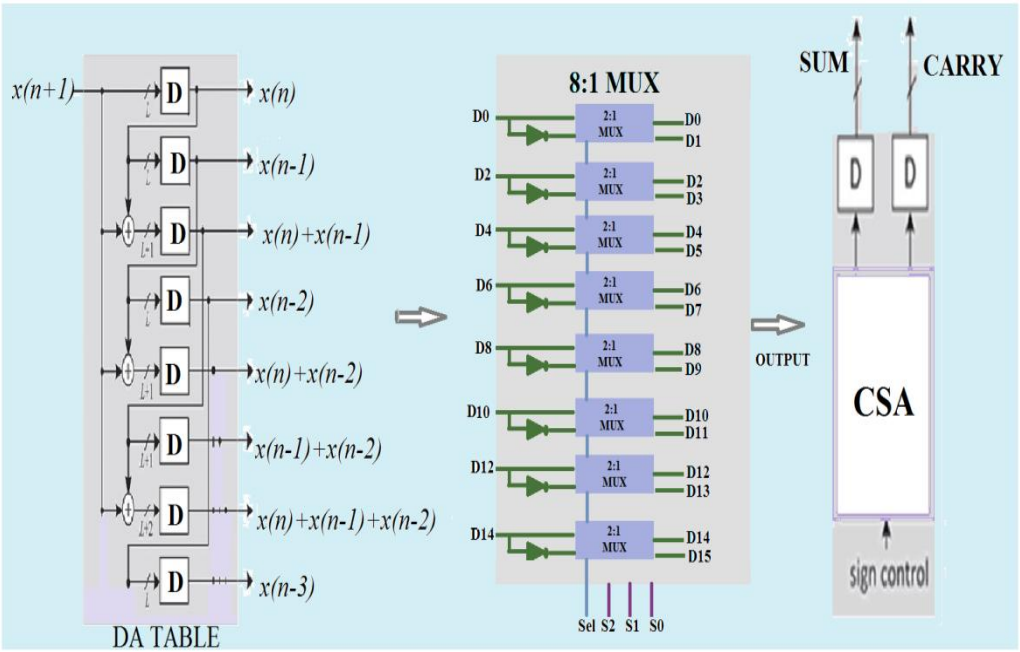


Fig. 2.Four-Point Inner Product Block

DA table forms the initial part of the 4-point inner product block, such that it consists of an array of registers .They are used for the purpose of storage of partial inner products y1.In the proposed approach, for the implementation of 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 structure, we can reduce the original DA-table structure by two times, which increases the area efficiency of the design twice. The proposed structure for the DA-table which makes use of eight delay elements is shown above.8-to-1 MUX is used to select the contents of the registers of the DA table. For the MUX the bit slices of weights A=w3lw2lw1lw0l are fed as control or select lines to draw the contents of the DA table. The control here is in LSB-to-MSB order. And then the output of it fed to the carry save accumulator.

The process of shift accumulation is done in the CSA block. The input of the block is from the 16-to-1 MUX. The bit slices are fed one after the other in the order LSB-to- MSB the CSA block .Here we can see that for MSB slices, negative of LUT output accumulation is very much necessary, so the LUT output is passed through XOR gates and sign control input. When MSB slice occurs the XOR gate produce 1's complement as the sign control bit is set to 1 and then the sum and carry are obtained after L clock cycles.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 9, September 2015**

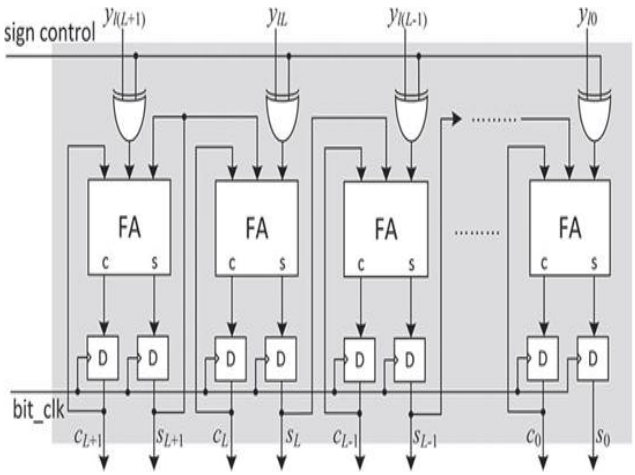


Fig. 3. Carry-Save Accumulator Block

B. *Weight Increment Block:*

It forms one of the important parts of the filter design structure. The weight increment unit for N=4 consist of four barrel shifters and four adder/subtractor cells. The structure of weight increment cell is shown in figure. It consists of barrel shifter, adder/subtractor, D-Flip flop and Word Parallel Bit-Serial Converter. It is used to update the weight of the four point inner product. The 8 bit outputs of the four point inner product block are given as inputs to this block. This helps in updating weight of the LUT's in distributed arithmetic.

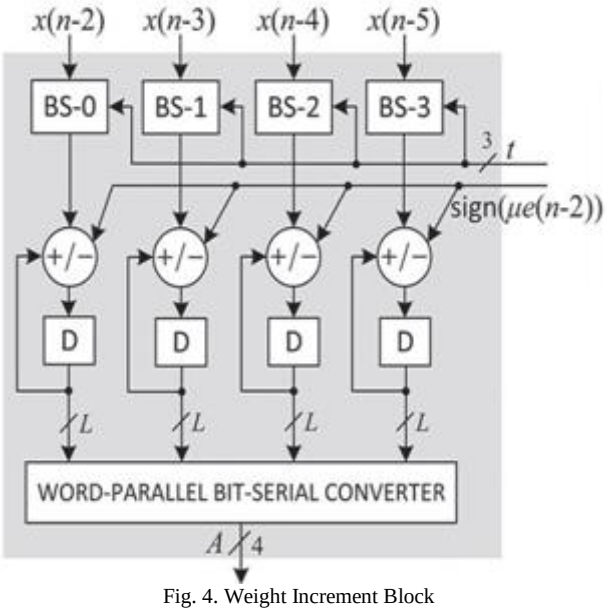


Fig. 4. Weight Increment Block

A barrel shifter is a combinational logic circuit with n data inputs, n data outputs, and a set of control inputs that specify how to shift the data between input and output. It is a digital circuit that can shift a data word by a specified number of bits in one clock cycle. The Control word't' for the working of the BS is produced by decoding the magnitude of error. This error that is decomposed is the difference between the desired output of the filter and the actual output produced by the filter. The logic for the selection for the control word't' for the BS is given in the table below.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 9, September 2015**

Table.1.Control Word Generation

| |
|---|
| if $r_6 = 1$ then $t =$ "000";<br>else if $r_5 = 1$ then $t =$ "001";<br>else if $r_4 = 1$ then $t =$ "010";<br>else if $r_3 = 1$ then $t =$ "011";<br>else if $r_2 = 1$ then $t =$ "100";<br>else if $r_1 = 1$ then $t =$ "101";<br>else if $r_0 = 1$ then $t =$ "110";<br>else then t = "111";<br><br>$r$=abs($\mu e(n-2)$)<br>$r_i$ : $i$th bit of 7-bit word $r$ |

Adder/subtractor Block is a digital circuit that is capable of adding or subtracting numbers. The circuit does the adding or subtracting process depending on a control signal. When Sign Bit = '0' the circuit perform addition .When Sign Bit = '1' the circuit perform subtraction.Word Parallel Bit Serial Converter is used to find convert parallel bit to serial

## IV. SIMULATION AND IMPLEMENTATION RESULTS

The simulation and the implementation results of the thesis are shown in this chapter. ISim and Xilinx ISE are the softwares used for the simulation. After simulation all the designed systems were implemented on the Xilinx Spartan 3E FPGA. The FPGA kit used for the implementation is Xilinx Spartan 3E (family), XC3S50 (device), ft 256 (Package), -4 (speed grade). The modules are modelled using VHDL in Xilinx ISE Design Suite 12.1 and the simulation of the design is performed using Modelsim SE 6.2c to verify the functionality of the design. Here a structural model is used for the coding purpose.

The figure given above shows the simulation result for the FIR filter. The DA table initiates the start of the filtering process. It is used to calculate the partial inner products. The inputs to this block are a clock signal, a reset and the 8-bit input signal'a'.When the res='0' and the clk ='1' the contents of the DA will be all zeros. Then when the res='1',the partial-product calculation is initiated, producing the partial inner products products by delaying and adding the input signal 'a', total of 8 outputs are produced of 10-bit size out of the DA table. This is then fed to the 8:1 MUX.The select line 'w' chooses the corresponding value from the MUX 'y1' and feeds it to CSA block.The respective output 'y2' is produced from CSA block. The filter output is subtracted from the desired signal'd', to produce the error signal 'e'. The error signal 'e',it is shifted to produce 'eint'.This signal is then decomposed to sign bit 'sign' by taking the MSB bit eint(7) and into magnitude 'mag' by taking all the bits of 'eint' excluding the MSB.The 'mag' is then passed through a control unit to produce the control signal 't' to drive the weight-increment block based on the select lines 'tcn'. The weight-increment block produces the weight 'w', by acting upon the inputs 'q0',q1','da(15)','da(16)' based on the signals 'sign' and 'tcn'.This then drives the 4-point inner product block to make the error minimum, ie , to make the output of the filter reach the desired value.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 9, September 2015**

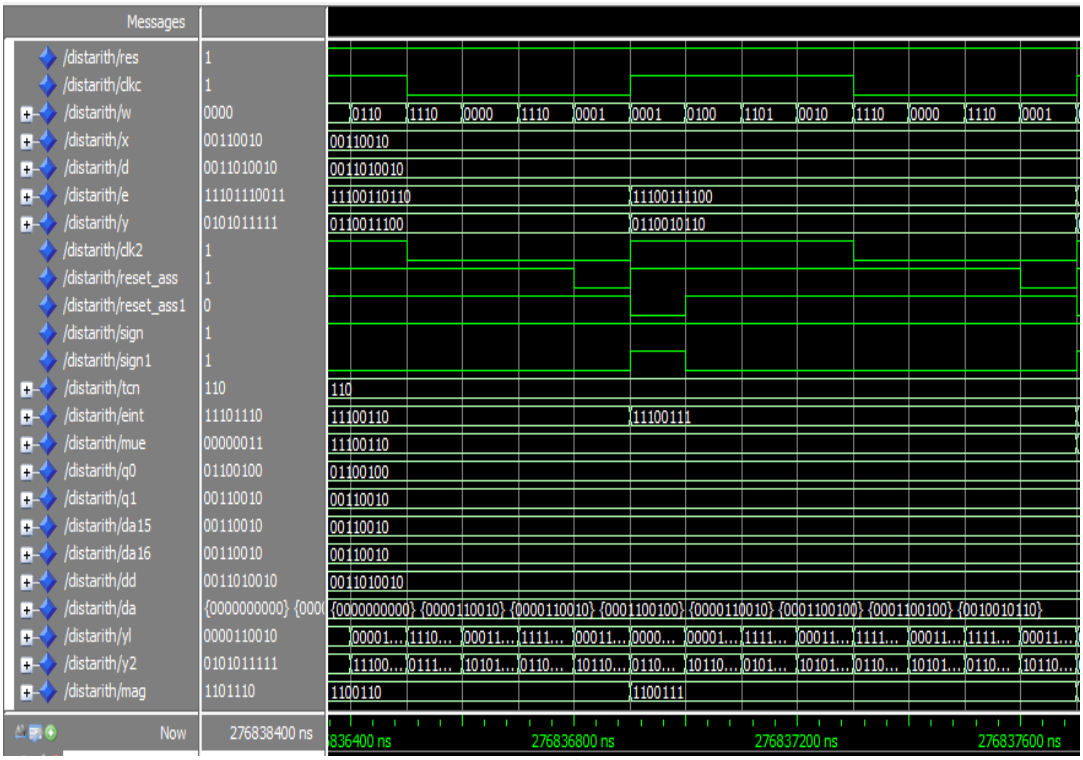


Fig. 5. Simulation Result ofAdaptive Fir Filter

The estimates of the power consumption for FIR filter are given in Table 3. From the results, the power consumption of the design is reduced significantly. For the attenuation in power consumption of proposed design, the system has the two separate clocks; slower for all computations except carry save accumulation. The carry save accumulation required separate fastest clock.

Table.2.Power Utilization of Filter

| Power summary: | I(mA) | P(mW) |
|---|---|---|
| Total estimated power consumption: | | 18 |
| | | |
| Vccint 1.20V: | 0 | 0 |
| Vccaux 2.50V: | 7 | 18 |
| Vcco25 2.50V: | 0 | 0 |
| | | |
| Clocks: | 0 | 0 |
| Inputs: | 0 | 0 |
| Logic: | 0 | 0 |
| Outputs: | | |
| Vcco25 | 0 | 0 |
| Signals: | 0 | 0 |
| | | |
| Quiescent Vccaux 2.50V: | 7 | 18 |

Area utilization of adaptive FIR filter is shown in table 3.Adaptive FIR Filter system will include relatively less number of look up tables, employed less number of adders to reduce required area of filter. The workload is reduced to re-computing only half of the LUT entries thereby significant area savings can be achieved by our proposed scheme

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 9, September 2015**

Table.3. Area Utilization of Filter

| Device Utilization Summary | | | | |
|---|---|---|---|---|
| **Logic Utilization** | **Used** | **Available** | **Utilization** | **Note(s)** |
| **Total Number Slice Registers** | 324 | 1,536 | 21% | |
| Number used as Flip Flops | 263 | | | |
| Number used as Latches | 61 | | | |
| Number of 4 input LUTs | 459 | 1,536 | 29% | |
| **Logic Distribution** | | | | |
| Number of occupied Slices | 330 | 768 | 42% | |
| Number of Slices containing only related logic | 330 | 330 | 100% | |
| Number of Slices containing unrelated logic | 0 | 330 | 0% | |
| **Total Number 4 input LUTs** | 527 | 1,536 | 34% | |
| Number used as logic | 459 | | | |
| Number used as a route-thru | 68 | | | |
| Number of bonded IOBs | 45 | 63 | 71% | |
| IOB Flip Flops | 18 | | | |
| IOB Latches | 14 | | | |
| Number of GCLKs | 3 | 8 | 37% | |
| **Total equivalent gate count for design** | 6,952 | | | |
| Additional JTAG gate count for IOBs | 2,160 | | | |

## V. CONCLUSION

Low-complexity design implementations have greater importance in efficient hardware implementations. In this brief architecture for the implementation of adaptive FIR filters is proposed. This design achieved low-complexity compared to existing DA-based implementations. By using the proposed structure, we can reduce the original DA-table structure by two times, which increases the area efficiency of the design twice. The proposed structure for the DA-table which makes use of eight delay elements. Thus the area and power has reduced considerably.We have implemented a carry-save accumulation scheme of signed partial inner products for the computation of filter output. From the synthesis results, we find that the proposed design consumes less power and less ADP over our previous DA-based FIR adaptive filter in average for filter length.

## REFERENCES

1. P. K. MEHER AND S. Y. PARK,"LOW-POWER, HIGH-THROUGHPUT, AND LOW-AREA ADAPTIVE FIR FILTER BASED ON DISTRIBUTED ARITHMETIC" *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS VOL. 60, NO. 6, JUNE 2013.*
2. R.Haimi-Cohen, H.Herzberg, and Y.Beery, "Delayed adaptive LMS filtering: Current results," in Proc.IEEE Int. Conf. Acoust., Speech, 1273–1276.
3. R.D.Poltmann, "Conversion of the delayed LMS algorithm into the LMS algorithm," IEEE Signal Process. Lett., vol. 2, p. 223, Dec. 1995
4. S.C.Douglas, Q. Zhu, and K. F. Smith, "A pipelined LMS adaptive FIR filter architecture without adaptive delay," IEEE Trans. Signal Process, vol. 46, pp. 775–779, Mar. 1998.
5. S.A.White, "Applications of distributed arithmetic to digital signal processing: A tutorial review," IEEE ASSP Mag., vol. 6, pp. 4–19, Jul. 1989.
6. 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., vol. 52, no. 7, pp. 1327–1337, Jul. 2005.
7. R.Guo and L.S.DeBrunner, "Two high performance adaptive filter implementation schemes using distributed arithmetic," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 58, no. 9, pp. 600–604, Sep. 2011.S.
8. R.Jayashri, H.Chitra, H.Kusuma, A. V. Pavitra, and V. Chandrakanth, "Memory based architecture to implement simplified block LMS algorithm on FPGA," in Proc. Int. Conf. Commun. Signal Process. (ICCSP), Feb. 10–12, 2011, pp. 179–183.
9. S.Baghel and R.Shaik, "FPGA implementation of fast block LMS adaptive filter using distributed arithmetic for high-throughput," in Proc. Int. Conf. Commun. Signal Process. (ICCSP), Feb. 10–12, 2011, pp. 443–447.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

*(An ISO 3297: 2007 Certified Organization)*

**Vol. 3, Issue 9, September 2015**

10. S.Baghel and R.Shaik, "Low power and less complex implementation of fast block LMS adaptive filter using distributed arithmetic," in Proc. IEEE Students Technol. Symp., Jan. 14–16, 2011, pp. 214–219.

## BIOGRAPHY

**Jesmin Joy** is an M-Tech scholar in VLSI and Embedded System in the Electronics and Communication Department, IIET, M.G.University. She received B-Tech degree in 2013 from M. G. University, Kottayam, Kerala. Her research interests are VLSI and HDL languages etc.