ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

LMS Algorithm and Distributed Arithmetic Based Adaptive FIR Filter with Low Area Complexity

Nilesh S. Satpute[1], Sanjay B. Tembhurne[2], Vipin S. Bhure[3]

M. Tech. PG Scholar VLSI Designing, Dept of Electronics and Communication, RTM Nagpur University, Nagpur, India[1]

M. Tech. Professor VLSI Designing, Dept of Electronics and Communication, RTM Nagpur University, Nagpur, India[2,3]

Abstract: In this manuscript, an adaptive FIR filter for high throughput, area and power efficient design will be introducing using distributed arithmetic (DA). DA is a bit serial computational action and uses equivalent concurrent realization of filtering weight update proposal for improving the throughput rate. As well as for high throughput rate and low area consumption the DA uses set of smaller dynamic parallel look up tables (LUTs). To reduced area requirement, sampling period and critical path, the conditional carry save accumulation of shift accumulator using full adder string circuitry will used in placed of conventional adder based shift accumulation. The Least mean square (LMS) algorithm is introduced to update weight and decline the mean square root error between desired and expected output. 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. The designed Adaptive FIR Filter system will include relatively less number of look up tables, employed half adders in replace of some full adders to reduce required area of filter and less number of multiplexer and thus required power consumption will be less.

Keyword: Adaptive filter, Distributed arithmetic (DA), least mean square (LMS) algorithm, LUTs, Inner Product unit.

I. INTRODUCTION

ADAPTIVE FIR filter is a system with a linear filter that has a transfer function managed by variable parameters and a means to adjust those parameters according to a suitable algorithm. Most adaptive FIR filters are digital filters only because of the difficulty faced of the optimization algorithms.

Why Adaptive filters?
Because some parameters of the desired processing operation are not known in advance or are varying. To implement the knowledge of Adaptive filtering and FIR filtering together for low power consumption with low area and high throughput making filtering system much better and convenient for Digital Signal System

Adaptive filters are widely used in several digital signal processing applications. The most usually used adaptive filter is the tapped-delay line finite impulse response (FIR) filter whose weights are updated by the famous Widrow–Hoff least mean square (LMS) algorithm. Because it has not only simple in nature but also it has satisfactory convergence performance [1]. The direct form configuration on the forward path of the FIR filter results in a long critical path due to an inner-product computation to obtain a filter output. Therefore, it is necessary to minimize the critical path of the structure so that the critical path could not beat the sampling period, when the input signal sampling rate has a high. In current years, without multiplier DA-based system [2] has gained for the significant popularity for its high-throughput processing potential and reliability, which result in cost-effective and area–time efficient computing structures. Hardware-capable DA-based design of adaptive filter has been suggested by Allred et al using two separate lookup tables (LUTs) for filtering and weight update. Author[3], have enhanced there system for filtering as well as weight updating by using only one lookup table. However, the system do not support high sampling rate for each new sample since they occupy several cycles for LUT updates. In a recent manuscript anticipated a resourceful design DA-based adaptive filter with very low alteration delay and with high-speed [4].
This designed based on DA and LMS algorithm for lesser area, power as well as very high-throughput pipelined realization of adaptive FIR filter with minimum adaptation time delay.

The designed system advantages are as follows.

1) By using a parallel LUT update the throughput rate is extensively improved.
2) Also throughput is significantly increased by concurrent implementation of filtering and weight updating.
3) In this, uses a conditional carry-save accumulation of signed partial inner products to reduce the sampling period instead of Conventional adder-based shift accumulation. The designed signed carry-save accumulation also helps to minimum the area complication of the designed filter.

DOI 10.17148/IJARCCE.2015.4677

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

4) By introducing a fast bit clock for carry-save accumulation and a much slower clock for all other computation, the reduction of power consumption is achieved.

II. DESIGN METHODOLOGY

LMS Algorithm

The LMS update algorithm is particularly simple if the variable filter has FIR tapped delay line in nature. Normally, after each sample, the FIR filter coefficients are adjusted as below:

$$W_{l,k+1} = 2\mu\, \epsilon_k x_{k-1} + W_{lk} \ldots\ldots.1$$

for μ is called the convergence factor.

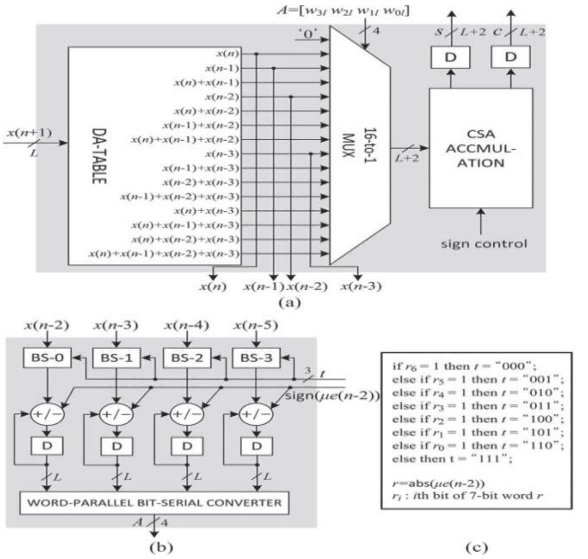
The LMS algorithm does not need that x values have any particular bond; therefore it can be used to adjust an FIR filter and a linear combiner. So the equation is given by:

$$W_{l,k+1} = 2\mu\, \epsilon_k x_{lk} + W_{lk} \ldots\ldots\ldots2$$

The effect of the LMS algorithm is at each time, k, to create a small change in each weight. The direction of the change is such that it would reduce the error value if it had been used at time k. The change in each weight magnitude depends on the associated x value, convergence factor μ and the error at time k. The output changed the most, as any change in values of weight magnitude. There should be no change in the weights, only at the time of error free system. The changing the weight makes no difference, if the associated value of x is zero [2].

III. PROPOSED DA-BASED ADAPTIVE FILTER STRUCTURE

The computation of adaptive filters of large orders needs to be decomposed into small adaptive filtering blocks since DA based implementation of inner product of long vectors requires a very large LUT [3]. Therefore, we describe here the proposed DA-based structures of small and large-order LMS adaptive filters separately in the two sections.



```
if r6 = 1 then t = "000";
else if r5 = 1 then t = "001";
else if r4 = 1 then t = "010";
else if r3 = 1 then t = "011";
else if r2 = 1 then t = "100";
else if r1 = 1 then t = "101";
else if r0 = 1 then t = "110";
else then t = "111";

r=abs(μe(n-2))
ri : ith bit of 7-bit word r
```

Figure 1.(a) Structure of the 4-PIPB. (b) Structure of the weight-increment block for N = 4. (c) Logic used for generation of control word t for the barrel shifter for L = 8.

In the LMS adaptive filter, in each cycle, wants to execute an inner-product computation which creates to the most of the critical path. For simplicity of presentation, let the inner product will be given by,

$$y = \sum_{k=0}^{N-1} w_k x_k \qquad \ldots\ldots\ldots(3)$$

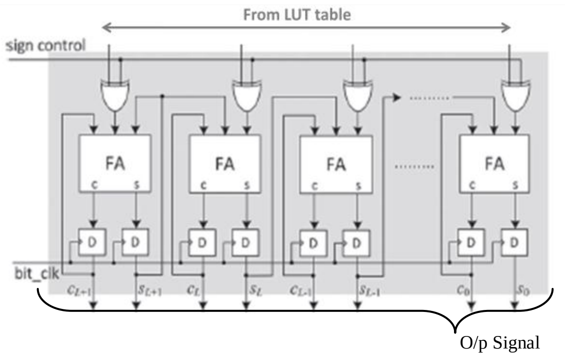


Figure 1(d). Structure of Carry Save Accumulator

A. Small-Order Adaptive Filter

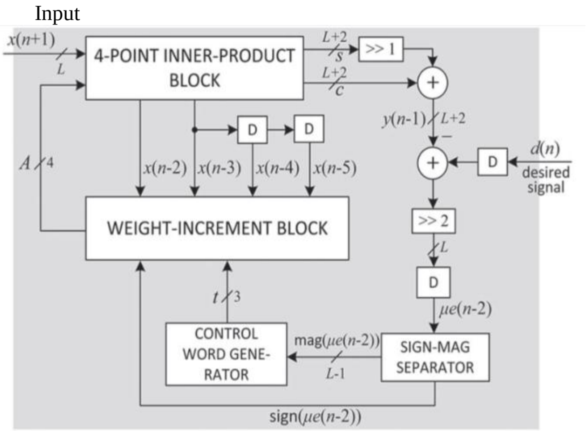


Figure 2. Block Diagram of Adaptive FIR Filter for order N=4

The above figure 4 shows block diagram of the Adaptive FIR Filter having order N=4 which is the combination of DA based 4-point inner product schematic 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.

The four-point inner-product block [shown in Fig. 1(a)] includes a DA table consisting of an array of 15 registers which stores the partial inner products yl for 0 < l ≤ 15 and a 16 : 1 multiplexor (MUX) to select the content of one of those registers. Bit slices of weights A = {w3l w2l w1l w0l} for 0 ≤ l ≤ L − 1 are transfer to the MUX as control in LSB-to-MSB order, and the output of the MUX is send to the carry-save accumulator. The carry-save accumulator shift accumulates all the partial inner products, after L bit cycles and calculates a sum word and a carry word of size (L + 2) bit each. The carry and sum words are shifted added with an input carry "1" to produce filter output which is subsequently subtracted to obtain the error e(n) from the desired output d(n).

DOI 10.17148/IJARCCE.2015.4677

As in the case in [3], all the bits of the error except the most significant one are ignored, such that multiplication of input x_k by the error is implemented by a right shift through the number of locations given by the number of leading zeros in the magnitude of the error. The error magnitude is then decoded to introduce the control word t for the barrel shifter. Figure 1(c) shows used logic for the control word t generation for barrel shifter. The convergence factor μ is usually taken to be O(1/N). We have taken μ =1/N. However, one can take μ as 2^i/N, where i is a small integer. In that case, to reduce the hardware complexity the number of shifts t is expand by i, and the input to the barrel shifters is pre-shifted by i locations. The weight-increment unit [shown in Fig. 1(b)] for N = 4 consists of four barrel shifters and four adder/subtractor cells.

The barrel shifter shifts the different input values x_k for k = 0, 1. . . N − 1 by appropriate number of locations (determined by the location of the most significant one in the estimated error). The barrel shifter yields the desired increments to be added with or subtracted from the current weights. The sign bit of the error is introduced as the control for adder/subtractor cells such that, when sign bit is one the BS output is subtracted from the content of the corresponding current value in the weight register and when sign bit is zero it will added to the content of the corresponding current value in the weight register.

B. Large-Order Adaptive Filter N=32

The inner-product computation of (3) can be decomposed into N/P (assuming that N = PQ) small adaptive filtering blocks1 of filter length P as

$$y = \sum_{k=0}^{P-1} w_k x_k + \sum_{k=P}^{2P-1} w_k x_k \ldots + \sum_{k=N-P}^{N-1} w_k x_k \quad \ldots\ldots\ldots\ldots(4)$$

From above equation (4), each of these P-point inner-product computation blocks will accordingly have a weight-increment unit to update P weights. The proposed structure designed for higher order N = 32 and P = 4. It consists of eight inner-product blocks of length P = 4, which is shown in Fig. 1(a). The (L + 2)-bit sums and carry produced by the eight blocks are added by two separate binary adder trees. Eight carry-in bits should be added to sum words which are output of eight 4-point inner-product blocks. Since the sum words are half of the carry words weight, two carry-in bits are put as input carry at the first level binary adder tree of carry words, which is corresponding to inclusion of eight carry-in bits to the sum words. Assuming that μ = 1/N, we truncate the eight LSBs of e(n) for N = 32 to make the word length of sign-magnitude separator be L bit.

IV. RESULT

A. Result states of 4-point inner product block

The figure 3 shows output states of 4-point inner product block and states of weight increment block.

The DA based 4-PIPB is mainly used to perform an internal computational part like addition, substraction, multiplication, ANDing, ORing etc. It consists with two main units; the DA table and carry save accumulator at the output which are connected by using 16:1 multiplexer shown in figure 1(a).

The input of the filter is internally given to the 4-PIPB i.e. first block DA table. The DA table is the bunch of number of LUTs, when 8 bit input given to the DA table it will compute all possible combinations and calculations and stored at LUTs. These values are called to perform filtering operations according to weight select lines of 16:1 mux.

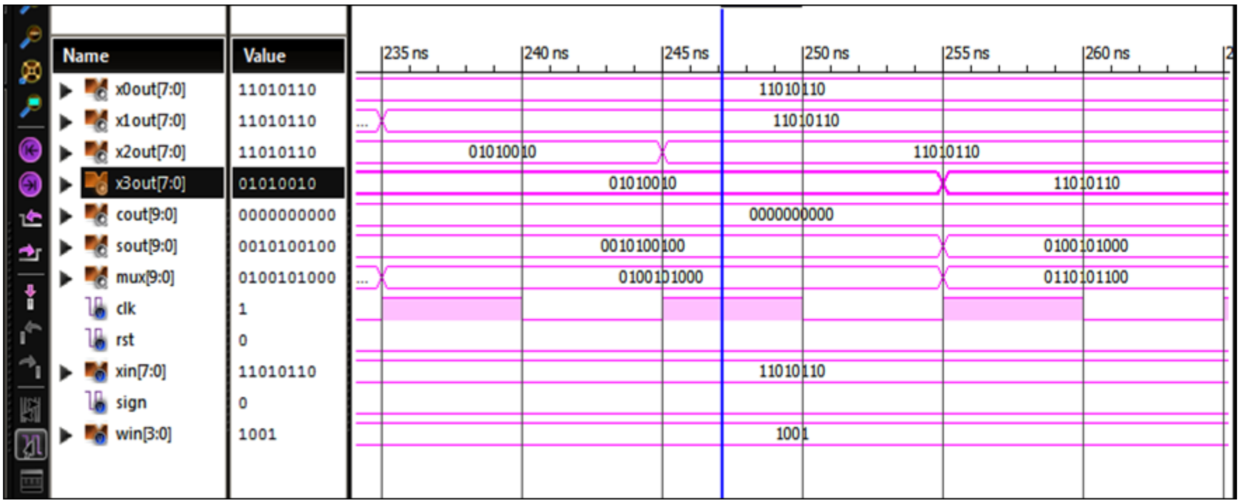


Figure 3. Output of 4-point Inner Product Unit

The weight select is decided by weight increment block shown in figure 1(b).

The output of the mux is given to the carry save accumulator (figure 1(d)) which is the combination of full- adder string.

The CSA has another one bit input known as sign control bit which is used to identify MSB slices appears as address.

DOI 10.17148/IJARCCE.2015.4677

B. Result states of Adaptive FIR Filter, Order N=32

The below figure 4(a) shows RTL View Black Box and figure 4(b) shows the final simulated result of Adaptive FIR Filter, order N=32 based on Distributed Arithmetic and LMS Algorithm.

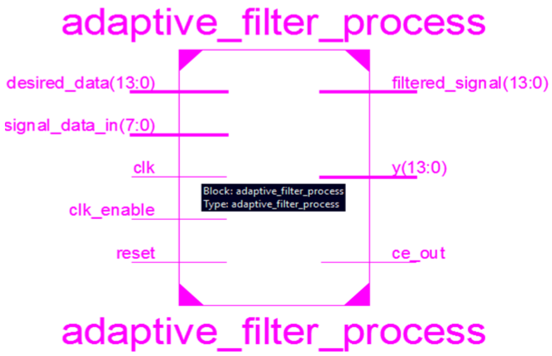


Figure 4(a). RTL View of AFIR Filter for order N=32

The AFIR Filter for order N-32 consists of eight units of 4-PIPB and WIB together. These blocks are combining by using half adders. The filter has two inputs; one is the 8 bit input signal and other one is desired response. During each cycle, the LMS algorithm computes a filter output and an error value that is equal to the difference between the current filter output and the desired response. In every training cycle the calculated error is then used to update the filter weights. The weights of LMS adaptive filter throughout the nth iteration are restructured according to the equations (1a). During first clock cycle, the 8 bit input signal is fed to the 4-PIPB-0, it will computes the sum and carry signals according to WIB-0 and another signal to the next 4-PIPB (i.e. 4-PIPB-1). Again the 4-PIPB-1 will computes the sum and carry output signals according to WIB-1 and one signal for next 4-PIPB. This cycle will continuing till 4-PIPB-7. After that we got eight sum output signals and eight carry output signals. These signals combine together using adders and finally we got filtered output of length L+5 shown in figure 6.

C. Result states of Adaptive FIR Filter, Order N=32on MATLAB

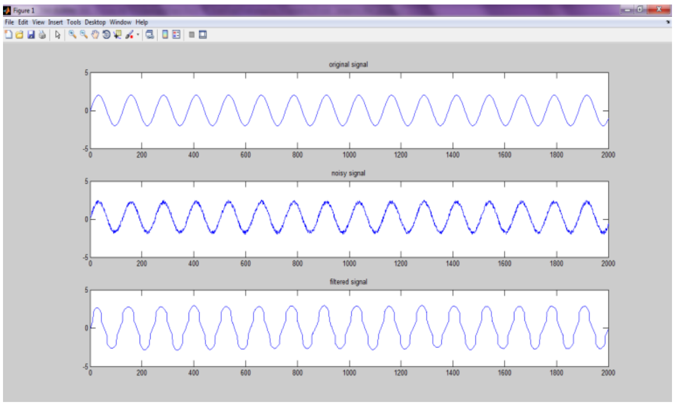
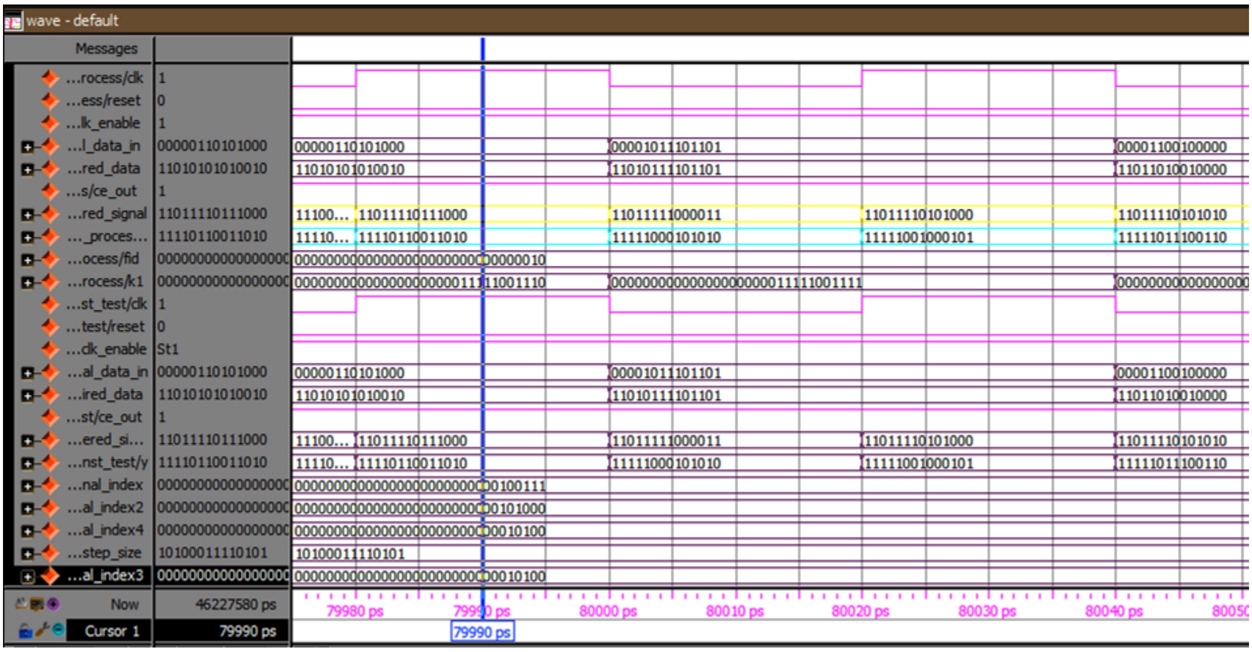


Figure 5. MATLAB final simulation



```
Messages
...rocess/clk      1
...ess/reset       0
...lk_enable       1
...l_data_in       00000110101000
...red_data        11010101010010
...s/ce_out        1
...red_signal      11011110111000
..._proces...      11110110011010
...ocess/fid       0000000000000000
...rocess/k1       0000000000000000
...st_test/clk     1
...test/reset      0
...clk_enable      St1
...al_data_in      00000110101000
...ired_data       11010101010010
...st/ce_out       1
...ered_si...      11011110111000
...nst_test/y      11110110011010
...nal_index       0000000000000000
...al_index2       0000000000000000
...al_index4       0000000000000000
...step_size       10100011110101
...al_index3       0000000000000000
Now                46227580 ps
Cursor 1           79990 ps
```

Figure 4(b). Output for AFIR Filter for order N=32

The above figure 5 shows MATLAB result simulation for same input given to the input of filter, which is separately process in MATLAB. I

In the above figure, first sine wave signal is the input signal given to the AFIR Filter. After that when we added some random noise into the first signal we got second noisy signal.

This noisy signal is then fed through the designed AFIR Filter, order N=32, we got third filtered signal at the output of filter.

DOI 10.17148/IJARCCE.2015.4677

IV. COMPARISON

Table-1 Comparing Different parameters of Proposed Methodology with existing

Parameters	EXISTING SYSTEMS			Proposed
	Meher & S.Y.Park	S. Y. Park	Aishwarya	
No. of Slice Registers	2991	2743	2098	2190
No. of LUT's	1430	1325	1201	1209
No. of Slice Flip-flops	2910	2743	2011	2190
No. of IOBs	94	60	87	54

V. CONCLUSION

We have suggested efficient pipeline architecture for smaller area and high throughput implementation DA based Adaptive FIR Filter. Throughput rate is significantly enhanced by using separate set of smaller dynamic LUT updates, equivalent concurrent processing of filtering operation and weight update operation. We have also proposed a conditional carry save accumulator to reduced area complexity, sampling period and critical path. From the analysis of synthesis result we found that the proposed filter design for N=32 consume 15% less area over previous DA-based Adaptive FIR Filter.

VI. FUTURE SCOPE

The designed AFIR Filter Distributed Arithmetic based system is limited to 8 bit data input system i.e. we cannot give more than 8 bit data at a time, which make filtering process little bit slow. In this, we designed filter system by pipelining architecture for smaller area and high throughput. So if we used pipelining architecture with parallel method it may works for large number of input bits with faster processing.

REFERENCES

[1] Sang Yoon Park, June 2013, "Low power, High Throughput, and Low-Area Adaptive FIR Filter Based on Distributed Arithmetic,"IEEE Transactions on circuits and systems-II: Express Briefs,Vol.60.

[2] G.Selvapriya, M.Mano, K.RekhaSwathiSri, PG Scholars, Mr S.Karthick, Assistant Professor (Sr. G), Department of ECE, Bannari Amman Institute of Technology, India, "High Throughput, Low Area, Low Power Distributed Arithmetic Formulation for Adaptive Filter", International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014.

[3] Aishwarya C PG scholar and Mr. Vijaybhaskar R Assist. Prof. Anna University, RegionalCentre, Coimbatore, "Enhanced Pipelined Architecture for Adaptive FIR Filter Based on Distributed Arithmetic," International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682 Vol.23, No23, March 2014.

[4] K. Jebin Roy, R. Ramya, "Low Power and Low Area Adaptive FIR Filter based on DA and LMS Algorithm" International Journal of Scientific and Research Publications, Volume 4, Issue 3, March 2014.

[5] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, Jul. 2005,"LMS adaptive filters using distributed arithmetic for high throughput," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 7, pp. 1327–1337.

[6] R. Guo and L. S. DeBrunner, Sep. 2011, "Two high-performance adaptive filter implementation schemes using distributed arithmetic," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 58, no. 9, pp. 600–604.

[7] K. Backiyalakshmi, PG Scholar, M. Raja, Assistant Professor, Veltech Multitech Dr. Rangarajan Dr. Sakunthala Engineering College "Design of FIR Filter Based on RLS Algorithm Using Filter Coefficient" ,International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 12, December 2013.

[8] M. Yazhini ,PG Scholar and R. Ramesh, Professor, Department of Electronics and Communication Engineering, Saveetha Engineering College, Tamil Nadu, 602105, India, "FIR Filter Implementation using Modified Distributed Arithmetic Architecture", Indian Journal of Science and Technology , Print ISSN: 0974-6846 , ISSN: 0974-5645, volume 6(5), May 2013.

[9] K. R. Borisagar, G. R. kulkarni "Simulation and Comparative Analysis of LMS and RLS Algorithms Using Real Time Speech Input Signal" GJRE, 2010.

[10] NJ Bershad, JCM Bermudez, "An Affine Combination of Two LMS Adaptive Filter Transient Mean-Squre Analysis" Signal Processing, IEEE Transactions, May 2008.

[11] K. Backiyalakshmi, PG Scholar, M. Raja, Assistant Professor, Veltech Multitech Dr. Rangarajan Dr. Sakunthala Engineering College "Design of FIR Filter Based on RLS Algorithm Using Filter Coefficient" ,International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 3, Issue 12, December 2013.

[12] Jyotsna Yadav, Mukesh Kumar, SHIATS, Allahabad, India, "Performance Analyis Of Lms Adaptive Fir Filter And Rls Adaptive Fir Filter For Noise Cancellation", Signal & Image Processing : An International Journal (SIPIJ) Vol.4, No.3, 4304,June 2013.