ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijareeie.com

Vol. 7, Issue 9, September 2018

Design of an Low Power and Area Efficient DA Based Fir Filter Using LMS Algorithm

M.K.Manu[1] , K.Sujatha[2], Dr. Kiran Bailey[3]

PG Student [Electronics], Dept. of ECE, BMS Engineering College, Bengaluru, India[1]

Associate Professor, Dept. of ECE, BMS Engineering College, Bengaluru, India[2]

Assistant Professor, Dept. of ECE, BMS Engineering College, Bengaluru, India[3]

ABSTRACT: This project presents a novel pipelined architecture for low-power, high-throughput, and low-area implementation of adaptive filter based on distributed arithmetic (DA). The throughput rate of the proposed design is significantly increased by parallel lookup table (LUT) update and concurrent implementation of filtering and weight-update operations. The conventional adder-based shift accumulation for DA-based inner-product computation is replaced by conditional signed carry-save accumulation in order to reduce the sampling period and area complexity. Reduction of power consumption is achieved in the proposed design by using a fast bit clock for carry-save accumulation but a much slower clock for all other operations. The least mean square (LMS) algorithm adaptation is functioned to update the weight and abate the mean square error between the assessed and chosen output. The weight increment block based adder and subtractor cells is exchanged by carry save adder in order to reduce area difficulty. The memory size can be reduced by decomposing the look up table (LUTs).

By the proposed method, area efficiency, low power and high throughput is achieved. It involves the same number of multiplexors, smaller LUT, and nearly half the number of adders compared to the existing DA-based design. From synthesis results, it is found that the proposed design consumes 13% less power and 29% less area-delay product (ADP) over our previous DA-based adaptive filter in average for filter lengths N = 16 and 32.Compared to the best of other existing designs, our proposed architecture provides 9.5 times less power and 4.6 times less ADP.

KEYWORDS: Adaptive Filter, Distributed Arithmetic (DA), Finite Impulse Response (FIR), Least Mean Square (LMS) Algorithm, Lookup table (LUT).

I.INTRODUCTION

Filters of some sort are essential to the operation of most electronic circuits. It is therefore in the interest of anyone involved in electronic circuit design to have the ability to develop filter circuits capable of meeting a given set of specifications. In circuit theory, a filter is an electrical network that alters the amplitude and/or phase characteristics of a signal with respect to frequency. Ideally, a filter will not add new frequencies to the input signal, nor will it change the component frequencies of that signal, but it will change the relative amplitudes of the various frequency components and/or their phase relationships. Filters are often used in electronic systems to emphasize signals in certain frequency ranges and reject signals in other frequency ranges. Such a filter has a gain which is dependent on signal frequency. The Least Mean Square (LMS) adaptive filter is the most popular and most widely used adaptive filter, not only because of its simplicity but also because of its satisfactory convergence performance. The direct form LMS adaptive filter involves a long critical path due to an inner product computation to obtain the filter output. The critical path is required to be reduced by pipelined implementation when it exceeds the desired sample period. Since the conventional LMS algorithm does not support pipelined implementation because of its recursive behavior, it is modified to a form called the delayed LMS (DLMS) algorithm, which allows pipelined implementation of the filter. A lot of work has been done to implement the DLMS algorithm in systolic architectures to increase the maximum usable frequency, but, they involve anadaptation delay of ~N cycles for filter length N, which is quite high for large order filters. With the completion of the project we have designed adaptive filter based in the frequency domain. This adaptive signal process will be applicable where the length of the unknown system's impulse response is long enough for practical Implementation of frequency domain adaptivefiltering.

DOI:10.15662/IJAREEIE.2018.0709011

II.REVIEW OF LMS ADAPTIVE ALGORITHM

A system and method has been developed to minimize the error signal and yield anoriginal signal that is fallowing a desired signal. The salient features of proposed system are:

Adaptive noise cancellation:

Adaptive filter is used in noise cancellation applications. The desired signal is combination of source signal and noise signal which is uncorrelated to the signal as shown in Fig. 1a.Filter takes a noise input and correlates with the noise in desired signal to obtain the actual signal. Input of a filter is a reference noise which is correlated with the noise in the desired signal. The error term e (n) obtained from the system is then used to cancel the noise in the original signal by using the LMS algorithm.

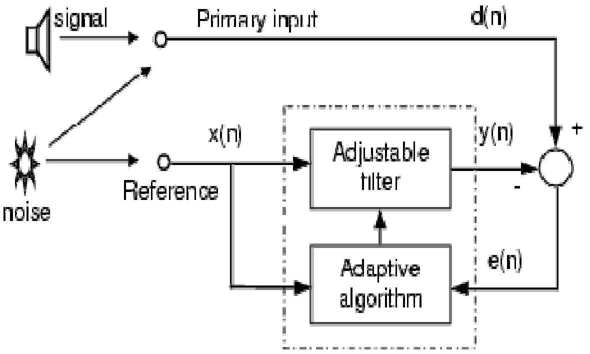


Figure 1a: Architecture block diagram of LMS fir filter with input source

LMS algorithm

Most widely use algorithm in adaptive filter is an LMS algorithm due to its simplicity. It doesn’t needs an extra mathematical calculation like matrix inversion nor correlation function. Mean Square Error (MSE) logic is used in LMS algorithm. It uses an input signal, step-size parameter, the subtraction of desired signal and filter output signal for calculating the updated filter coefficients.

LMS Equation

Based on the filter taps and input response the equation will be obtained for number of iterations. The equation to updated tap weight w (n) using the input signal x (n) and the desired response d (n) with step size μ is shown in equation (1).

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\, \mathbf{x}(n)[\, d(n) - \mathbf{x}(n)\mathbf{w}(n)\,] \ldots\ldots. \ (1)$$

Whereas μ is the step size, hence the filter output is the sum of the product of tap weights and input signal.

$$y\,(n) = \mathbf{x}\,(n) * \mathbf{w}\,(n) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots... \ (2a)$$

Error signal e (n) is defined as the subtraction of the desired signal and the filter response signal.

$$e\,(n) = d\,(n) - y\,(n) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.... \ (2b)$$

So, Equation (1) can be further written in terms of the error signal and the tap weights.

$$\mathbf{w}\,(n+1) = \mathbf{w}\,(n) + \mu\, \mathbf{x}\,(n)\, e\,(n) \ldots\ldots\ldots\ldots\ldots\ldots.. \ (3)$$

The formula for the LMS algorithm is shown in equation (3). As illustrated in the equation, each updated tap weight needs the current tap weight and the current errorsignal obtained from the desired response after subtraction. The algorithm doesn’trequire depth knowledge of the whole cross correlation vector or autocorrelation matrixdoesn’t require matrix computations.

DOI:10.15662/IJAREEIE.2018.0709011

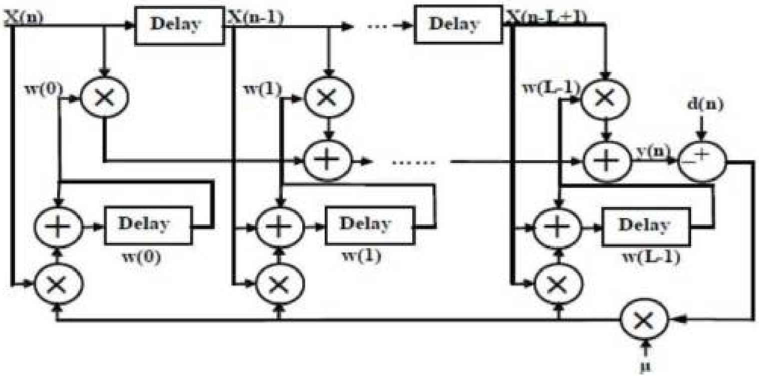


Figure 1b: Structural view of FIR filter with weights using LMS algorithm

The LMS algorithm uses a structure of an FIR filter. The structural view of FIR filter is shown in Fig. 1b. From the figure b, filter as two main components those are L delay registers and weight update blocks. The Unit Delay Registers are made of simple D Flip-Flops. And each Weight Update component consists of a multiplier, adder and a buffer to store the new updated weights of the filter coefficient.

III.PROPOSED DA-BASED APPROACH FOR INNER PRODUCT COMPUTATION

In each cycle, the LMS adaptive filter needs to perform an inner critical path product computation which given by

$$y = \sum_{k=0}^{N-1} r_k \, . \, s_k \quad (4)$$

Where $r_k$and s_k for $0 \leq k \leq N-1$ form the N – point vectors corresponding to the current weights and most recent $N-1$ input respectively. Let us assume L be the bit width of the weight, every component of the vector weight may be expressed in 2's complement representation

$$r_k = -r_{k0} + \sum_{l=1}^{L-1} r_{kl} \, . \, 2^{-l} (5)$$

Where w_{kl} denotes the lth bit of r_k . Substituting (5), we can write (4) in an expanded form

$$y = -\sum_{k=0}^{N-1} s_k \, . \, r_{k0} + \sum_{k=0}^{N-1} s_k \, . [\sum_{l=1}^{L-1} r_{kl} \, . \, 2^{-l}] (6)$$

To convert the sum-of-product form of (4) into a distributed form, the order of summations over the indices and in (6) can be interchanged to have

(7)

And the inner product given by (7) can be computed as

$$y = [\sum_{l-1}^{L-1} 2^{-l} \, . \, y_l] - y_0 \, , \, y_l = \sum_{k=0}^{N-1} s_k \, . \, r_{kl} \ (8)$$

Meanwhile any element of the N-point bit sequence{ r_{kl} for 0 <k<N-1 } can either be 1 or 0, the partial sum y1 for l= 0,1,…,L-1, can have 2^N possible values. If all the 2^N possible values sum are precomputed and stored in aLUT, the partial sum can be read out from the LUT usingthe bit sequence { r_{kl} } as address bits for computing the innerproduct.

$$y = -\sum_{k=0}^{N-1} s_k \, . \, r_{k0} + \sum_{l=0}^{L-1} 2^{-l} \, . [\sum_{k=0}^{N-1} s_k \, . \, r_{kl}]$$

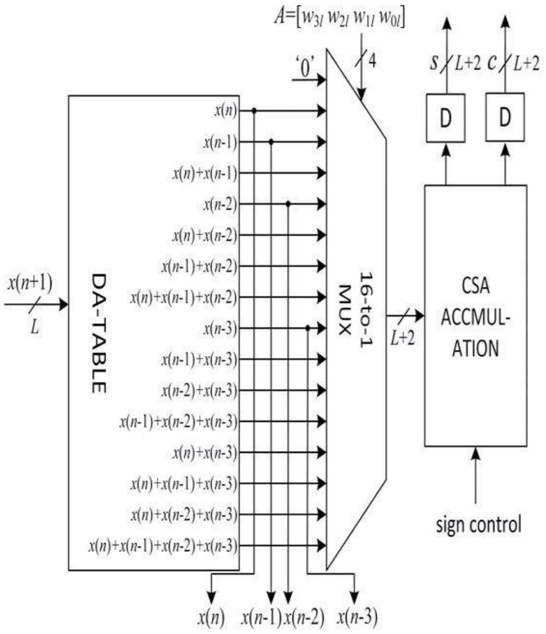


Figure 1: DA-based implementation of four point inner product

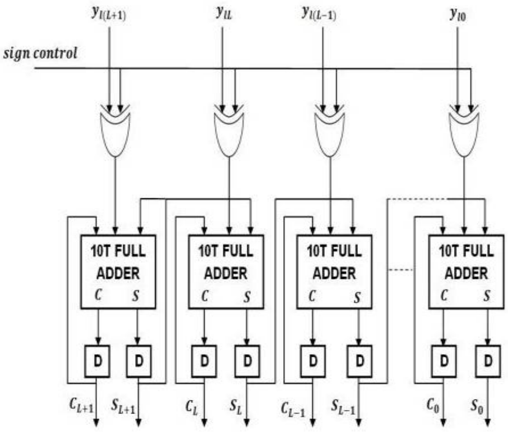


Fig 2: Carry save implementation of shift accumulation

The inner product of (8) can therefore be calculated in cyclesof carry save implementation of shift accumulation, followed by LUT-read operations corresponding to number of bitslices { r_{kl} } for$0 \leq l \leq L-1$is shown in Fig. 1. Since the carry save implementation of shift accumulation in Fig. 2required more area and power consumption.

DOI:10.15662/IJAREEIE.2018.0709011

The carry save implementation of shift accumulation based fulladder is design by using one bit-full asshown in Fig. 2. The bit slices of vector are fed one after thenext in the LSB to the MSB order to the carry saveaccumulator. Finally, the sum and carry output of the carrysave accumulator is obtained after clock cycle are required tobe added by a final adder. The content of the LUT locationcan be expressed as,

$$c_k = \sum_{j=0}^{N-1} x_j \, . \, k_j$$

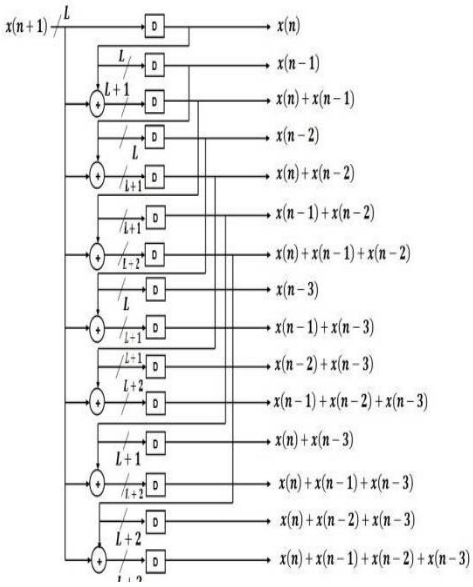


Figure 3: Distributed arithmetic table.

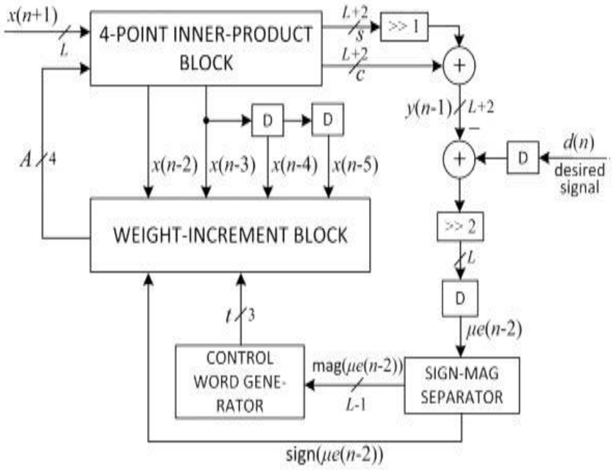


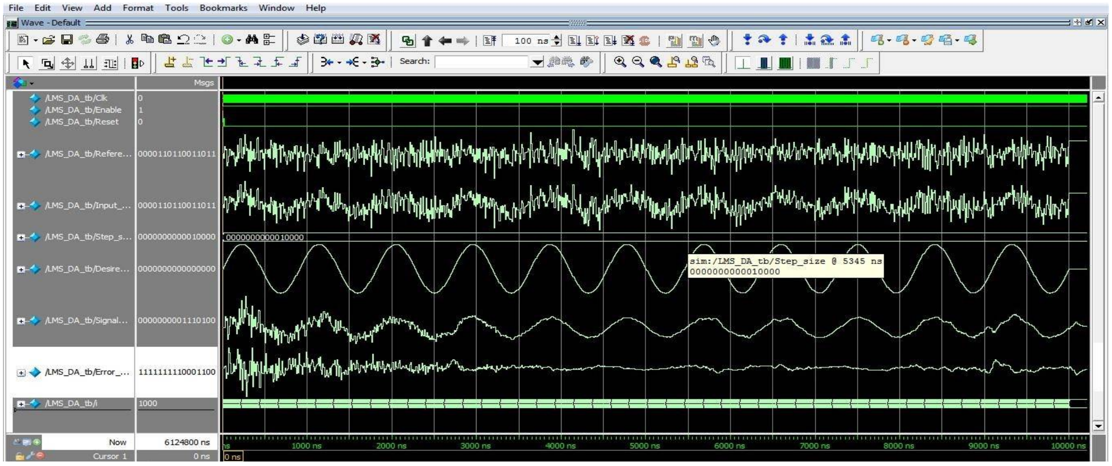
Figure 4: Proposed structure of DA-based LMS adaptive filterlength N=4.

DA table for N=4 is shown in Fig. 3. DA table contains only15 registers to store the pre calculated sums of input words. InDA table, seven new values of are computed by sevenadders in parallel.

IV. RESULT AND DISCUSSION

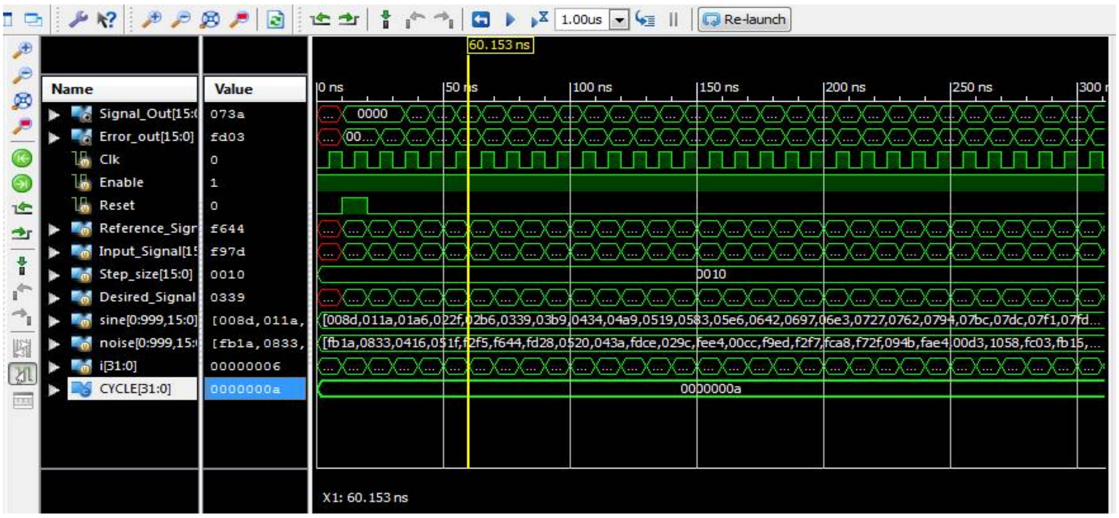
In this chapter we will discuss about results in brief using some of captured simulation results from two different tools modelsim and Xilinx ,The modelsim is a rare tool not important in terms of simulating the Verilog coded RTL but for this project we needed to observe the analog output what we are getting back i.e. original output which one retrieving from the received signal , so visually we can analysis how much we are retrieving back .In terms of analog and digital representation both are same results for the designed system.

In the below figure you can see the simulation result in term of analog waveform which was obtained by simulating the Verilog RTL code in the modelsim Tool. Anyhow has per the analysis we can't achieve 100% accuracy in term of getting back the original signal because all system will have their own internal noise margin , that to in DSP processing units it's very hard to achieve 90% accuracy has we seen in the industry perspective. Here In the below figure if you see carefully the output signal fallowing the input signal almost but don't have exact replica has input and also you can observe the error fluctuation and error removed from the received signal in the below figure.


/LMS_DA_tb/Clk 0
/LMS_DA_tb/Enable 1
/LMS_DA_tb/Reset 0
/LMS_DA_tb/Refere... 0000110110011011
/LMS_DA_tb/Input_... 0000110110011011
/LMS_DA_tb/Step_s... 0000000000010000
/LMS_DA_tb/Desire... 0000000000000000
/LMS_DA_tb/Signal... 0000000001110100
/LMS_DA_tb/Error_... 1111111110001100
/LMS_DA_tb/i 1000
Now 6124800 ns
Cursor 1 0 ns

sim:/LMS_DA_tb/Step_size @ 5345 ns
0000000000010000

Figure 5: The figure given above shows the simulation result in terms of analog for the designed FIR filter.


Name	Value
Signal_Out[15:0]	073a
Error_out[15:0]	fd03
Clk	0
Enable	1
Reset	0
Reference_Signal	f644
Input_Signal[15	f97d
Step_size[15:0]	0010
Desired_Signal	0339
sine[0:999,15:0]	[008d,011a,
noise[0:999,15:0	[fb1a,0833,
i[31:0]	00000006
CYCLE[31:0]	0000000a

[008d,011a,01a6,022f,02b6,0339,03b9,0434,04a9,0519,0583,05e6,0642,0697,06e3,0727,0762,0794,07bc,07dc,07f1,07fd...
[fb1a,0833,0416,051f,f2f5,f644,fd28,0520,043a,fdce,029c,fee4,00cc,f9ed,f2f7,fca8,f72f,094b,fae4,00d3,1058,fc03,fb16,...
X1: 60.153 ns

Figure 6: The figure given above shows the simulation result at the different time cycle.

V.CONCLUSION

We have suggested an efficient architecture for low power, high throughput, and low area implementation of DA based adaptive filter. Throughput rate is significantly enhanced by parallel LUT update and concurrent processing of filtering operation and weight update operation. We have also proposed a carry save accumulation scheme of signed partial inner products for the computation of filter output and also modified in weight increment block. By this way it utilizes low area, low power consumption and the throughput of the filter rates increases irrespective of the filter length. From the synthesis results, we find that the proposed design consumes 13% less power and 29% less area consumption over conventional FIR adaptive filter in average for filter lengths N = 4. Compared to the best of other existing designs, our proposed architecture provides 9.5 times less power and 4.6 times less area.

So here we proposed the digital filter using FIR, so in future it can be implement in IIR also using both Frequency and Time domain based on the domain and requirement where it can be applicable for specific work and also the number of logics level slices and transistors used can be reduce in future because has day by day we are moving to lower technology nodes, in terms of power and area also can reduce if it is implemented in very lower node technology , apart from technology in terms of algorithm we can improve the design by doing the lots of iteration using various available algorithms in the industry perspective.

REFERENCES

1. P. K. Meher and Y. Park,”low-power, highthroughput, and low-areaadaptive 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. 2008.
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. 2008.
5. S.a.white, “applications of distributed arithmetic to digital signal processing: a tutorial review,” ieee assp mag., vol. 6, pp. 4–19, jul.2012.
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. 2011.
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, 2013, pp. 179–183. Low power and area efficient fir filter using adaptive lms algorithm bmsce, mtech, ece page 14.
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, 2015, pp. 443– 447.
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, 2016, pp. 214–219.
11. Sang Yoon Park, Member, IEEE, and Pramod Kumar Meher, SeniorMember, IEEE “Low-Power, High-Throughput, and Low-AreaAdaptive FIR Filter Based on Distributed Arithmetic” IEEETransactions On Circuits And Systems—Ii: Express Briefs, Vol.60, No. 6 June 2013.
12. A.B.Diggikar, S.S.Ardhapurkar “ ImplementingFSM to control Adaptive Filter for Noise Cancelling in Speech signal”, Proc. of the International Conferenceon Advanced Computing and Communication Technologies, 452-457,2011.