FIXED POINT LEAST MEAN SQUARE ADAPTIVE FINITE IMPULSE RESPONSE FILTER

Patan.Aleem khan[1], V.JeanShilpa[2]

[1]*Department of Electronics Communication& Engineering, B.S.A.U, Chennai, (India)*

[2] *Assistant professor, Dept. of Electronics communication& engineering, B.S.A.U, Chennai, (India)*

ABSTRACT

Fir filters is highly used in Digital communication & Signal processing applications Digital radio receivers, Downconverts Software Radio.in this project I present an efficient architecture for the implementation of a delayed least mean square adaptive filter.Sofor achievingthis lower adaptation-delay and area-delay-power efficient implementation, I use a novel partial product generator and propose a strategy for optimized balanced pipelining across the time-consuming combinational blocks of structure to the proposed architecture. This paper implements the less area delay and less energy delay compare to existing one. From this one I am saving the 20% in ADP and 9% in EDP compare to the existing one in this paper I am using 8 bit filter length this is normally uses in various application like digital signal processing.

Keywords - Adaptive Filters, Circuit Optimization, Fixed-Point, Least Mean Square Algorithm

I. INTRODUCTION

The basic operation of the adaptive filter is LMS whereused to mimize a desired filter by finding the filter coefficients that relate to producing the least mean squares of the error signalThe least mean square (LMS) adaptive filter is the most popular and most widely used this one so not only because of its simplicity but also because for its satisfactory convergence performance.LMS algorithms is a class of adaptive filterThe direct-form LMS adaptive filter involves a long critical path due to an inner-product computation to obtain the filter output. In this paper existing one will take so much time to get the filter output.so I went for the delayed LMS adaptive filter which it will reduce the operation.

The critical path is required to be reduced by pipelined implementation when it exceeds the desired sample period. Asthe conventional LMS algorithms donot support pipelined implementation because of its recursive behavior .Therefore suitable pipelined implementation should be used in further architecture to reduce the area and it will have the high throughput & delay.

For the reason I implemented the DLMS adaptive filterit is same as least mean square but structure required delay is very low. Then it will allow the pipelined structure to reduce the critical path when it exceeds the sample period of the pipelined structure

Therefore I go through basic of fir filter which having linearity of phase and stability in frequency response and having constant group delay. It describes an FIR filter of length K. Theny[n] of the (i.e., (1) reads as “equation 1”) represented as

$$y[n] = \sum_{k=0}^{K-1} a_k x_{[n-k]}$$

x and y are the input and transformed data.

a_k will be the set of constant coefficients of the filter.

(K-1) is the order of the FIR filter.

An FIR filter can also be characterized by its number of taps (K), which is the order incremented by the one. Therefore the transfer function A(z) will be the FIR filter of the(i.e., (2) reads as "equation 2") is expressed as follows:

$$A(z) = \frac{Y(z)}{X(z)}$$

$$A(z) = \sum_{k=0}^{K-1} a_k z^{-k} = a_0 + a_1 z^{-1} + \ldots + a_{(K-1)} z^{-(K-1)}$$

Therefore FIR filter is also called an all-zero filter because the frequency response is only determined by the zeros in the z-transform.

FIR filters are mostly preferred because of its linear phase characteristic and stability. Therefore IIR filters can be used in applications that require sharp cut-off or narrow band filters and where linear phase is not required because FIR filters require much higher order implementations than IIR filters for a similar performance.

II. OPTIMIZATION OF FIXED POINT LMS ADAPTIVE FILTER

In this paper discuss the fixed-point implementation and optimization of the proposed DLMS adaptive filter. A bit level pruning of the adder tree is also proposed to reduce the hardware complexity without noticeable degradation of steady state MSE.

Then the weights of LMS adaptive filter during the *n*th iteration are used according to the following (i.e.; read as equation 3")is expressed as

$$wn+1 = wn+ \mu \cdot en \cdot xn$$

Therefore the input vector x*n*and the weight vector w*n*at the *n*th iteration are expressed as

$$xn= [xn, xn-1. . . xn-N+1]T$$

and

$$wn= [wn(0),wn(1), . . . ,wn(N - 1)]T ,$$

dnis the desired response, ynis the filter output, and en denotes the error computed during the nth iteration. Whereμis the step-size, and N is the number of weights used in the LMS adaptive filter.

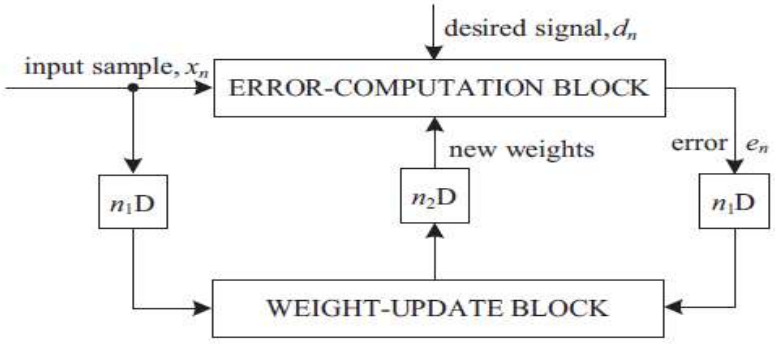

Fig.1: Modified Delayed Lms Adaptive Filter

Then the weight-update equation for DLMS adaptive filter of the (i.e., read as "equation 4") is expressed as

$$wn+1 = wn+ \mu \cdot en-m \cdot xn-m.$$

Then the adaptation delay of m cycles amounts to the delay introduced by the whole of adaptive filter structure consisting of finite impulse response (FIR) filtering and the weight-update process.

Then the weight update equation for modified DLMS algorithm is given as

$$en-n1= dn-n1-yn-n1$$

$$\& \; Yn= wTn-n2 \cdot xn.$$

We notice that during the weight update the error with *n*1 delays wereused. Then the filtering unit uses the weights delayed by *n*2 cycles. The modified DLMS algorithm decouples computations of the error-computation block and the weight-update block and allows us to perform optimal pipelining by feed forward cut-set retiming of both these sections separately to

less the number of pipeline stages and adaptation delay.

III. ERROR-COMPUTATION BLOCK

It having*N* number of 2-b partial product generators (PPG) corresponding to *N* multipliers and a cluster of *L*/2 binary adder trees, followed by a single shift–add tree.

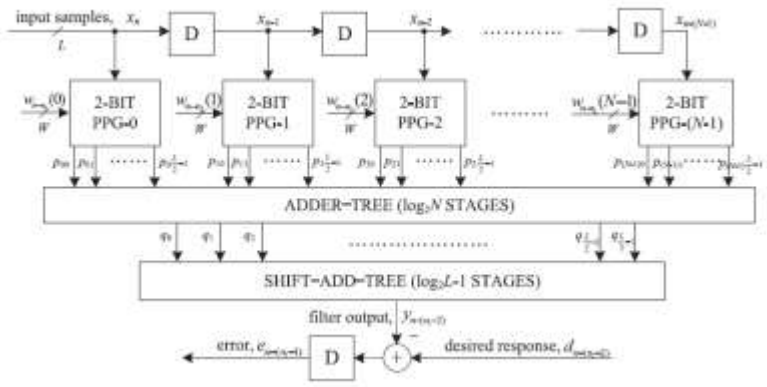

Fig.2: Error Computation Block

IV. WEIGHT UPDATED BLOCK

Therefore It will performs *N* multiply-accumulate operations of the form *(μ ×e) ×xi + wi*to update *N* filter weights.

The step size μ is taken as a negative power of 2 to realize the multiplication with recently available error only by a shift operation. Each of the MAC units therefore performs the multiplication of the shifted value of error with the delayed input samples *xi* followed by the additions with the corresponding previously weight values *wi*. Then All the *N* multiplications for the MAC operations were performed by NPPGs and to *N* shift– add trees.

This will leads to substantial reduction for the adder complexity. The final outputs of MAC units contains the desired updated weights to be used as inputs to the error-computation block as well as the weight-update block for the next iteration

Each of the PPGs generates *L*/2 partial products corresponding to the product of the recently shifted error value $\mu \times e$ with *L*/2,the number of 2-b digits of the input word *xi* , wr. Since the scaled error ($\mu \times e$) is multiplied with the entire *N* delayed input values in the weight-update block then this sub expression can be shared across all the multipliers as well.

.

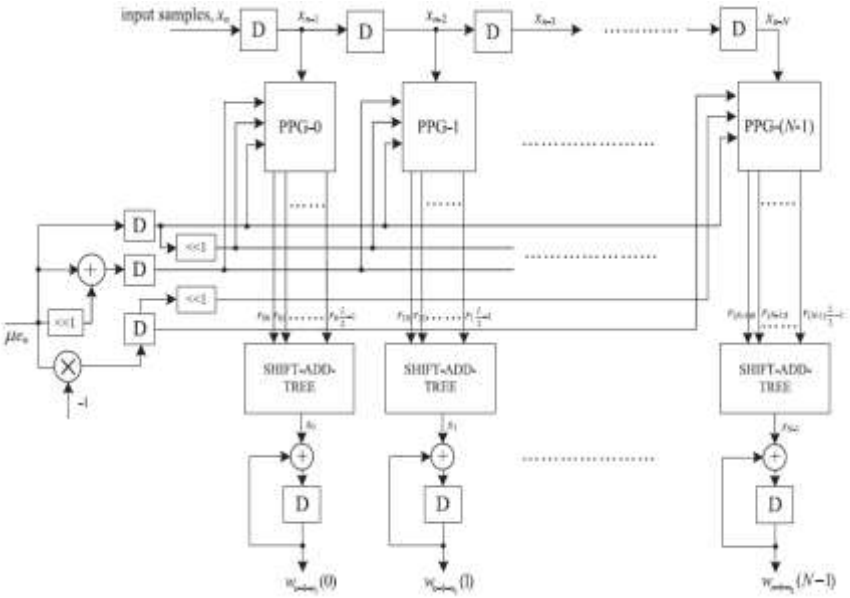

Fig.3: Weight Updated Block

V. RESULTS

In this paper comparison of the both existing and proposed system. Where no of LUT (look up tables) will be increased and the no of gates should be reduced compared to existing one output.

Then the simulation output will be given 8 digit binary values where It have the reset and clock pin and inputs and filtered output.

VI. PROPOSED OUTPUT

Where reset=1 the filtered output will be remains 0 then clock will be same the input signal will be xn and desired signal dn and the output will yn at 90 ns.

When reset=0 then the clock will be same the input signal will be xn and desired signal dn and the output will yn at 90 ns.

Device utilization of the adaptive fir filter in the existing one. In that number of gate count are high compare to proposed one.so that we are going for the better delay to reduce the computation tine in the pipelined architecture.

The results are shown in the figures are from simulation output and the Xilinx output and executed in the hardware part of the Spartan 3E board.

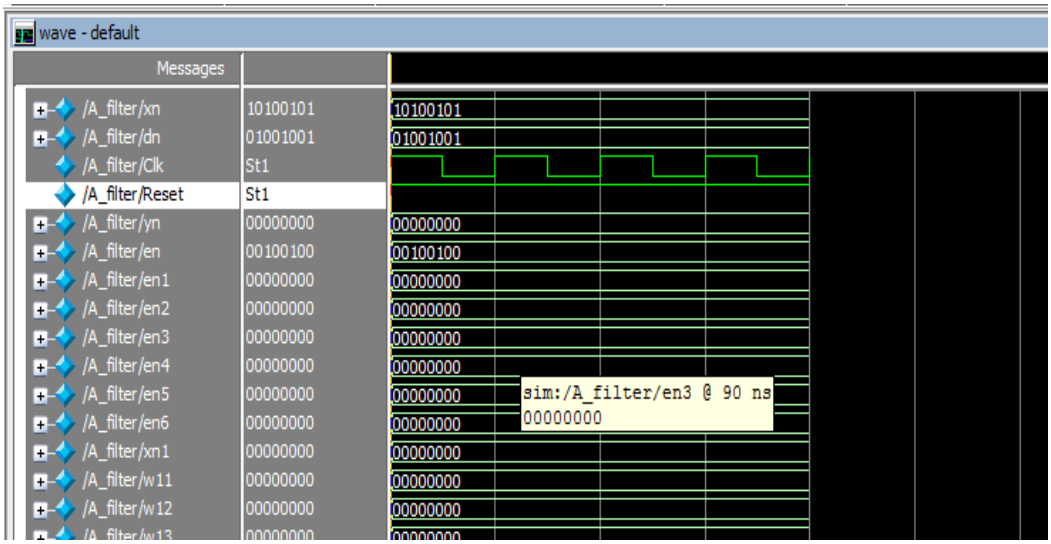

Fig.4: Proposed Wave Form Output

When reset=0 then the clock will be same the input signal will be xn and desired signal dn and the output will yn at 90 ns where xn=10100101 dn=01001001 then yn=00000000 then it will change periodically. Where "equation 3" is expressed as

$$yn=wn.xn,$$

$$wn+1 = wn+ \mu \cdot en \cdot xn$$

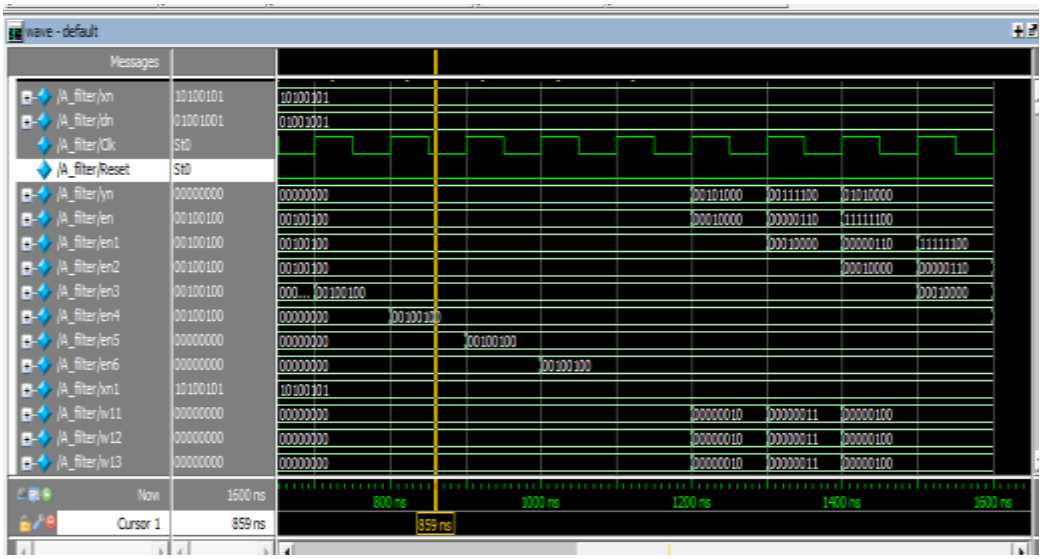

Fig. 4a: Proposed wave form output

Where en will change for next combination of input in the proposed system. Therefore the tabular column of compared the existing and proposed outputs

$$en= dn- ynyn= wTn\cdot xn$$

Comparison table of the existing and proposed outputs in the Xilinx software using Spartan 3e board get the results

6.1 Existing Output

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	132	7,168	1%	
Number of 4 input LUTs	435	7,168	6%	
Logic Distribution				
Number of occupied Slices	281	3,584	7%	
Number of Slices containing only related logic	281	281	100%	
Number of Slices containing unrelated logic	0	281	0%	
Total Number of 4 input LUTs	452	7,168	6%	
Number used as logic	435			
Number used as a route-thru	17			
Number of bonded IOBs	26	141	18%	
IOB Flip Flops	8			
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	5,295			
Additional JTAG gate count for IOBs	1,248			

TABLE 1

In the given table the total no of gates in the design is more compared to proposed one so they are using DLMS algorithm in the adaptive filter. Then getting the ouput from hardware called Spartan 3E

6.2 Proposed Output

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	143	7,168	1%	
Number of 4 input LUTs	435	7,168	6%	
Logic Distribution				
Number of occupied Slices	289	3,584	8%	
Number of Slices containing only related logic	289	289	100%	
Number of Slices containing unrelated logic	0	289	0%	
Total Number of 4 input LUTs	453	7,168	6%	
Number used as logic	435			
Number used as a route-thru	18			
Number of bonded IOBs	26	141	18%	
IOB Flip Flops	8			
Number of GCLKs	1	8	12%	
Total equivalent gate count for design	4,990			
Additional JTAG gate count for IOBs	1,248			

TABLE 2

In RTL schematic there is having the input desired signal and weight update block and they are giving the clock signal has 1 then it will set. Where reset will be 0 the operation of the process will start to execute.

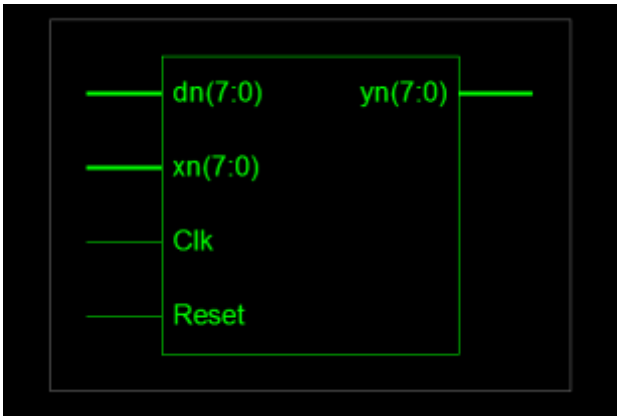

Fig.4c: RTL SCHEMATIC

In error computation block. It having structure adder trees where it performs the operation in the error computation and it have the 4 no of LUT's have been present .Therefore the gates in design will be reduced then they are having weight update block to reduce the computation time

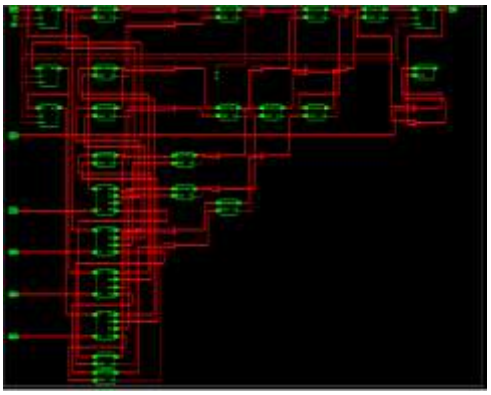

Fig.4d: ERROR COMPUTATION BLOCK

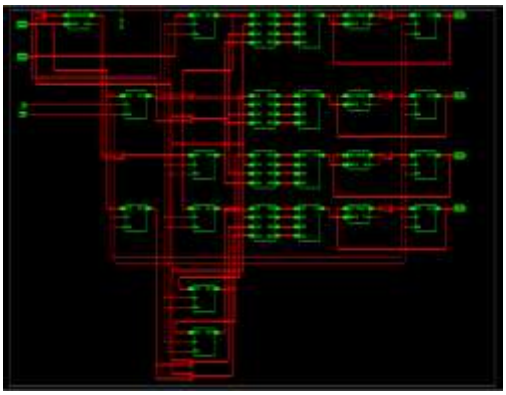

Fig. 4e: Weight Update Block

Then fig 4e represents the schematic flow of the weight update block getting from the Xilinx software and executed in Spartan 3E board.

VII. CONCLUSION

In this paper the advancement of performance will be reduced the adaptation delay and power consumption as well it will reduce the critical path to support high sampling rates. We found that the adaptation delay will provide significant saving of ADP and EDP compared to the existing one. The performance factor of the proposed one increased by 8% more than the existing one.

In future work we will modify the proposed system by reducing the Area and delay of the design.

REFERENCES

BOOKS

[1] B. Widrow and S. D. Stearns Adaptive Signal Processing Englewood Cliffs NJUSA: Prentice-Hall, 1985.

[2] S. Haykin and B. WidrowLeast-Mean-Square Adaptive Filters Hoboken NJ USA: Wiley, 2003

JOURNAL PAPERS

[3] M. D. Meyer and D. P. Agrawal, "A modular pipelined implementation of a delayed LMS transversal adaptive filter," in Proc. IEEE Int. Symp. Circuits Syst., May 1990, pp. 1943–1946.

[4] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," IEEE Trans. Acoust., Speech, Signal Process., vol. 37, no. 9, pp. 1397–1405, Sep. 1989.

[5] G. Long, F. Ling, and J. G. Proakis, "Corrections to 'The LMS algorithm with delayed coefficient adaptation'," IEEE Trans. Signal Process., vol. 40, no. 1, pp. 230–232, Jan. 1992.

[6] S. Ramanathan and V. Visvanathan, "A systolic architecture fo00000000000r LMS adaptive is filtering with minimal adaptation delay," in Proc. Int. Conf. Very Large Scale Integer. (VLSI) Design, Jan. 1996, pp. 286–289.

BIOGRAPHICAL NOTES

Mr. Patan. Aleem khan is presently pursuing M.tech final year in Electronics Communication & Engineering Department (specialization in VLSI&Embedded systems) from B.S.Abdur Rahman University Chennai, Tamilnadu, India

Mrs. V.JeanShilpais working as an Assistant Professor inElectronics Communication & EngineeringDepartment, B.S.Abdur Rahman UniversityChennai and presently pursuing Ph. D. from B.S.Abdur Rahman University, Chennai, Tamilnadu, India