

An Efficient Fixed Point DLMS Adaptive Filter For Denoising

^[1]Neethu Mary Raju, ^[2]Asha A S

^[1] PG Student [VLSI & ES] ^[2] Assistant professor,

Department of ECE, TKM Institute of Technology, Kollam

^[1]neethumaryraju@gmail.com ^[2]ashaprasobh@gmail.com

Abstract: The fixed point DLMS adaptive filter is employed for image processing applications and typical for image denoising application. Adaptive filter adapts the filter coefficients according to an optimization algorithm. Because of its simplicity and satisfactory convergence performance LMS (Least Mean Square) algorithm is used. It does not support pipelined implementation because of its recursive behaviour. It is modified to a form called the Delayed LMS (DLMS) algorithm, which allows pipelined implementation of the filter. For achieving lower adaptation delay as well as area and delay efficiency, partial product generators (PPG) and ripple carry adder for adding scheme is used. Adaptation delay can be reduced by decoupling the computations at the error computation block and weight update block. It also emphasis on fixed point issues as location of radix point choice of word length and quantization at various stages of computation. It is most popular for real-time adaptive system.

Index Terms—Adaptive filter, fixed point arithmetic, Least Mean Square (LMS) algorithm, image processing, image denoising

I. INTRODUCTION

Digital signal and image processing applications widely use the digital adaptive filters. An adaptive filter has transfer function controlled by variable parameters adjust themselves according to an optimization algorithm. Most of the adaptive filters are digital filters because of the complexity of the optimization algorithms. The filtering process is used in DSP and DIP real time applications. It can remove the noise from the original signal or image. The filter architecture helps to increase the system speed and to improve the system performance level. Adaptive filters have wide range of applications such as mobile phones and other communication devices, camcorders, digital cameras, medical monitoring equipments, etc.

Adaptive filtering includes two operations namely the filtering process as well as the adaptation process. Through filtering process, digital filter generates an output signal, by using an input data signal. At the same time adaptation process will adjust the coefficients of the filter for minimizing the desired cost function. Adaptive filters use different structures and algorithms based on different applications. Least Mean Square (LMS) algorithm provides better convergence performance and simple calculation. Inorder to track the desired filter output, the filter weights will be updated using the error information. The following mathematical model describes the LMS algorithm.

$$y_n = w_n^T \cdot x_n \quad (1)$$

$$e_n = d_n - y_n \quad (2)$$

$$w_{n+1} = w_n + \mu \cdot e_n \cdot x_n \quad (3)$$

Where y_n is the filter output, w_n^T is the filter weights in transposed form, x_n is the filter inputs usually measured by sensors, d_n is the desired filter output, e_n is the error signal and μ is the step size which is used for controlling the stability and the rate of convergence. The LMS algorithm cannot support pipelining because of its recursive behaviour, so DLMS algorithm. The DLMS algorithm is a pipelined representation of LMS algorithm which uses registers along with filter and error feedback path. The delayed LMS weight update equation is as follows:

$$w_{n+1} = w_n + \mu \cdot e_{n-D} \cdot x_{n-D} \quad (4)$$

where D represents the delay lines considering the pipeline architectures in DLMS adaptive filters.

II. LITERATURE SURVEY

Different systolic architectures are used to implement the DLMS algorithm to increase the maximum usable frequency [12], [11]. But if the filter length is N, then its adaptation delay increases approximately N cycles, which is quit high for large order filters. Also, the adaptation delay increases convergence performance degrades. A modified

systolic architecture have proposed by Visvanathan [6] to reduce the adaptation delay. For achieving a lower adaptation delay with the critical path of one MAC operation Van and Feng [10] have proposed a systolic architecture, where they have used relatively large processing elements (PEs). In order to limit the critical path to the maximum of one addition time Ting [7] have proposed a fine-grained pipelined design, which supports high sampling frequency. But it involves a lot of area overhead for pipelining and higher power consumption, due to its large number of pipeline latches. Further Meher and Park have proposed a 2bit multiplication cell, and used to minimize the critical path and silicon area without increasing the number of adaptation delays [4], [5].

III. PROPOSED SYSTEM

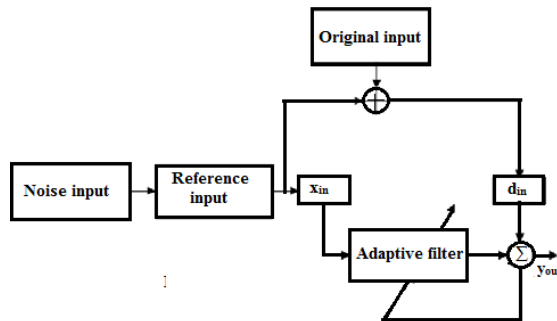


Fig. 1. Adaptive Noise Canceller

The goal is to produce denoised signal from the primary input. The basic block diagram of adaptive noise cancellation is shown in Fig. 1.

A. Salt and pepper noise source

The primary signal d_{in} consists of the superposition of noise signal and the original signal. The reference signal x_{in} is noise signal measured at the noise source. The original signal and the reference signal are the image and the noise image. The noise can be of different types Gaussian noise, uniform noise, Speckle Noise, Brownian Noise, salt and paper etc. Here considering the salt and paper noise because adaptive filter designed for $N=8$. Salt and pepper noise is also called intensity spikes as it's an impulse type noise. It has two possible values, the pepper noise is 0 and for salt noise 255 for an 8 bit image. Data transmission, malfunctioning of pixel elements in the faulty memory locations, camera sensors or timing errors in the digitization process etc lead to the salt and pepper noise. For reducing the error signal by using the inputs information and error values, the filter weights $w_0, w_1, w_2,$ and w_3 are updated, where the step size, $\mu=0.5$. The output signal y_{out} is filtered output from the fixed point DLMS adaptive filter.

B. DLMS Adaptive filter

The DLMS adaptive filter operations can be divided as Error computation block and Weight update block in order to reduce the adaptation delay.

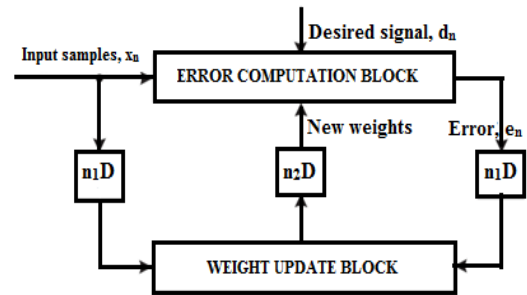


Fig. 2. DLMS Adaptive Filter

C. Pipelined Structure of the Error Computation Block:

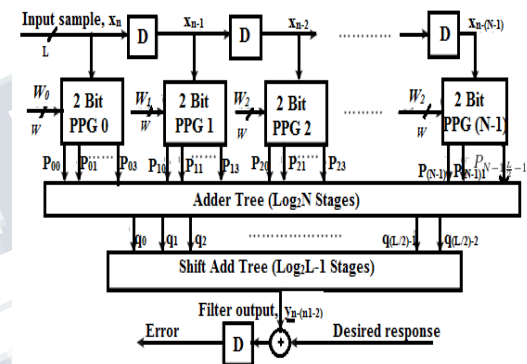


Fig. 3. Error Computation Block

The arrangement for error computation unit of an N-tap DLMS adaptive filter. It consists of N number of 2-bit partial product generators (PPG) and $\log_2 N$ stages followed by $\log_2 L - 1$ stages of shift add tree where $L = N/2$, where $N = 8, 16, 32$.

Structure of PPG: The structure of single PPG block is shown in Fig. 4. It composed of $L/2$ number of 2 to 3 decoders and AND/OR cells (AOC). Each of the 2 to 3 decoder module takes a 2-bit digit $(u_1 u_0)$ as input and then it will produces three outputs $b_0 = u_0 u_1, b_1 = u_0 \bar{u}_1$ and $b_2 = \bar{u}_0 u_1$ such that $b_0 = 1$ for $(u_1 u_0) = 1, b_1 = 1$ for $(u_1 u_0) = 2,$ and $b_2 = 1$ for $(u_1 u_0) = 3$. The decoder output b_0, b_1 and b_2 along with sign extended 2's complement representation of weight coefficients $w, 2w$ and $3w$ are fed to an AOC. For considering sign at the most significant digit (MSD) of the input samples at the partial product, i.e., $(u_{L-1} u_{L-2})$ of the input sample, the AOC $(L/2) - 1$ is fed with $w, 2w,$ and $-w$.

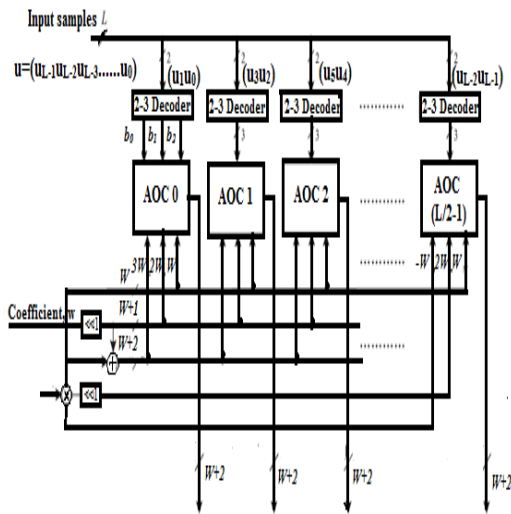


Fig. 4. Structure of PPG

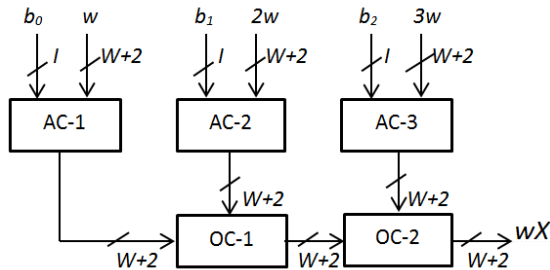


Fig. 5. Structure of AOC

Structure of AOCs: Fig. 5 shows the structure and function of an AOC. Each AOC consists of three AND cells and two OR cells. Each AND cell takes a single bit input b and an n -bit input. The other inputs of all the n AND gates are fed with the single bit input b . A pair of bits in the same bit position in is fed to the same OR gate. Corresponding to the decimal values as 1, 2, and 3 of the 2 bit input (u_1u_0) , respectively the output of an AOC is w , $2w$ and $3w$ is produced. Multiplication of input operand w with a 2 bit digit (u_1u_0) is done by using the decoder along with the AOC performs.

Structure of Adder Tree: Fig. 6 shows the structure and function of an AOC. The partial products of each PPG is shifted and add separately to obtain the product values and then added all the N product values to compute the desired inner product. But, the output obtained from the shift add operation followed by partial product increases the word length, and consequently increases the adder size of $N - 1$ additions of the product values. In order to avoid increase in word size of the adders, add all the N partial products on the same place value from all the N PPGs by one adder tree. Then all the $L/2$ partial products generated by each of the N

PPGs are thus added by $L/2$ binary adder trees. The outputs from the $L/2$ adder trees are then added by a shift add tree according to their place value.

D. Pipelined Structure of the Weight Update Block:

The structure for the weight update block is shown in Fig. 7. In weight update block the input is given to the PPG unit. The

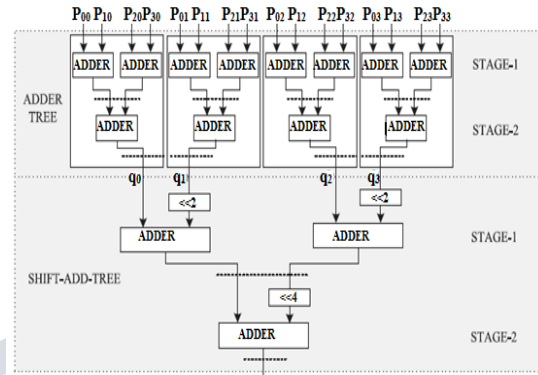


Fig. 6. Adder Structure

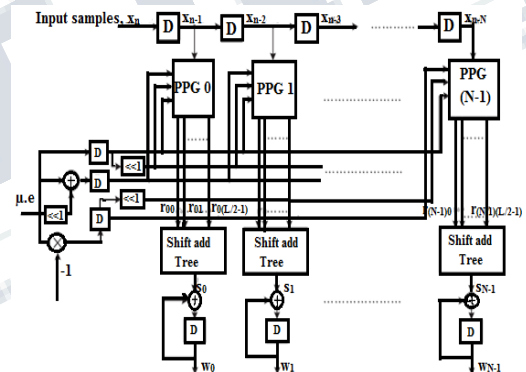


Fig. 7. Weight Update Block

N input it is multiplied by step size as in the form $(\mu.e).x_i + w_i$ to update the weight in the PPG unit. The PPG unit simple multiplies the input bits and the $\mu.e$. In the MSB part of the input is multiplied by negative in order to consider the sign bit at the MSB. It is then shifted and produces the output. The output from PPG is again given to MAC unit, where the PPG out is add and delayed and feedback it back. Each of the MAC units therefore performs the multiplication with the delayed input samples x_i and the shifted value of error followed by the shift additions with the corresponding old weight values w_i . The multiplication operations of the MAC operations are performed by N PPGs, followed by N shift add trees. The scaled error $(\mu.e)$ is multiplied with all the N delayed input values in the weight update block and then this subexpression will be shared across all the multipliers as well. This helps to reduce the complexity of the adder. The desired updated weights are constituted by the final output of

MAC units. Then it will be given as the input to the error computation block as well as the weight update block for the next iteration.

E. Salt and pepper Noise removal

The salt and pepper noise in the image can be removed using Fixed point DLMS Adaptive filter. This can be done such that the ASCII values of the original image and the noise image is produced. This values are read by the Fixed point DLMS adaptive filter and produces the approximate output, y_{out} as well as corresponding weight updates are produced as in Fig. 1.

IV. FIXED POINT DESIGN CONSIDERATIONS, ADDER TREE OPTIMIZATION

A. Fixed Point Design Considerations

For a fixed point implementation, the choice of word lengths and radix points for input samples, weights and internal signals need to be decided. Fig. 8 shows the fixed point representation of a binary number.

Let (X, X_i) be a fixed point representation of binary number where X is the word length and X_i is the integer length. The word length and location of radix point of x_n and w_n need to be predetermined by the hardware designer. By considering the design constraints as desired accuracy and hardware

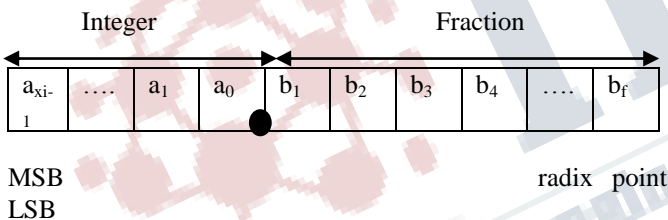


Fig. 8. Fixed point representation of binary numbers

TABLE I

Fixed Point Representation of Signals

Signal Name	Fixed Point Representation
x	(L, L_i)
w	(W, W_i)
p	$W+2, W_i+2$
q	$W+2+\log_2 N, W_i+2+\log_2 N$
y,d,e	$W, W_i+L_i+\log_2 N$
μ, e	W, W_i
r	$W+2, W_i+2$
s	W, W_i

Complexity. Assuming (L, L_i) and (W, W_i) , respectively as the representations of input signals and filter weights then all other signals can be decided as the Table I. Three times the value of input coefficients will be the output of PPG block. Thus, add two more bits to the word length and to the integer length of the coefficients to avoid overflow.

B. Adder Tree Optimization

y_n can be pruned for reducing the computation complexity and for the optimization of area, delay and power complexity. The pruning optimization done at adder tree and shift add tree at the computation of filter output. The partial products generated by the PPG unit, for $W=8$ in which each row of the dot diagram contains 10 dots. Each set of partial products of the same weight values contains four terms, for $N=4$. The final sum in product without truncation should be 18 bit. However, only use MSB 8 bit in the final sum, and the rest 10 bit in LSB are finally discarded.

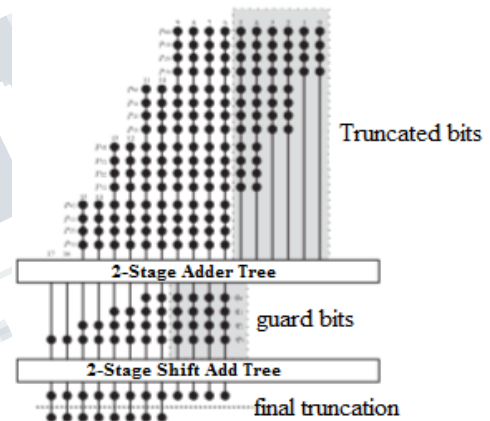


Fig. 9. Dot diagram for optimization of the adder tree in case of $N=4, L=8$ and $W=8$

V. EXPERIMENTAL RESULTS

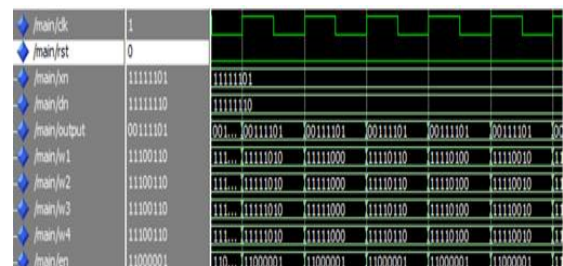


Fig. 10. Simulated waveform of DLMS Adaptive filter

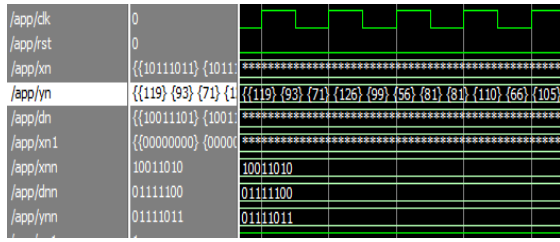


Fig. 11. Simulated output of adaptive noise canceller

A. Fixed point DLMS adaptive filter

The simulation of Fixed point DLMS adaptive filter as shown in Fig. 10.

B. Fixed point DLMS adaptive filter for image as input

The adaptive noise cancelled output using fixed point DLMS adaptive filter is shown in Fig. 11. The original signal, the reference signal and the output are the image, the noise image and the denoised image.

C. Image Denoising

The denoised image is obtained from the original image by suppressing noise from a noise contaminated version of the image as in Fig 12.

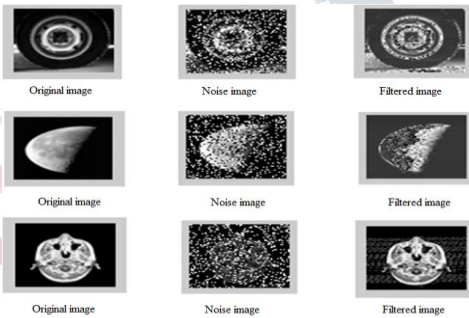


Fig. 12. Filtered output

D. Area Delay Comparison

**TABLE II
COMPARISON OF AREA AND DELAY**

Method	Area			Memory in kilobyte	Delay
	LUT	Gate count	Slices	Size	Gate delay
LMS Adaptive filter	117	8606	504	198096	12.979
Fixed point DLMS Adaptive filter	217	1805	217	141000	8.948

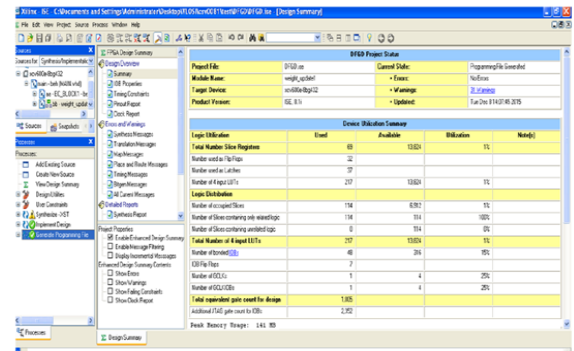


Fig. 13. Area of the DLMS Adaptive filter

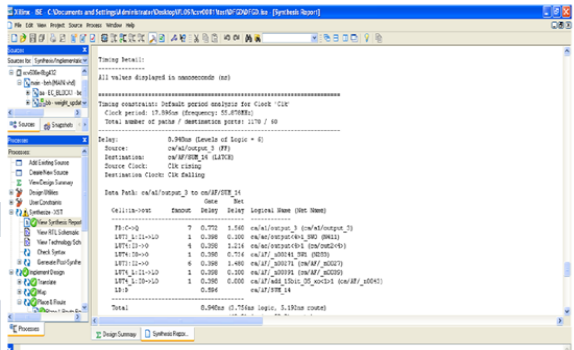


Fig. 14. Delay of the DLMS Adaptive filter

The comparison of area and delay between the existing LMS adaptive filter and proposed delayed LMS adaptive filter is given Table II.

VI. CONCLUSION

Digital adaptive filters are widely used for image noise cancellation applications. Adaptive filter adapts the filter coefficients according to an optimization algorithm. The cost function, which is used to determine the filter coefficients, is minimized using adaptive process. The squared error between output and a primary signal of an adaptive filter is minimized by adjusting its coefficient. The adaptive filter architecture is to increase the system speed and to improve the system performance level. A novel PPG for general multiplications and inner product computation by common sub expression sharing is designed for getting area and delay efficiency. Besides, an efficient ripple carry addition scheme for inner product computation that reduces the adaptation delay significantly in order to achieve faster convergence performance and reduced critical path. Apart from this, a strategy for optimized balanced pipelining across the time consuming blocks of the structure reduces the adaptation delay. The fixed point consideration reduces hardware complexity without noticeable degradation of steady state error. The fixed point DLMS adaptive can be used for efficient denoising purpose.

REFERENCES

- [1] Pramod Kumar Meher and SangYoon Park, "Area-Delay-Power Efficient Fixed Point LMS Adaptive Filter With Low Adaptation-Delay", in Proc. IEEE Int. Midwest Symp. Circuits Syst., Feb. 2014.
- [2] S. Moya Fargana and Mahendran, "Critical-Path Analysis and Low Complexity Implementation of the LMS Adaptive Algorithm", in Proc. IEEE Int. Symp. Circuit Syst., Feb 2014, pp. 110-114.
- [3] Farheen Fathima and P.Anuradha, "VLSI Implementation of Efficient Fixed-Point LMS Adaptive Filter with Low Adaptation Delay", in Proc. ICRACVESMay2014, pp. 137-142.
- [4] Hesam Ariyadoost and Yousef S. Kavian, "Two Dimensional Systolic Adaptive DLMS FIR Filters for Image Processing on FPGA", in Proc. IEEE Int. Symp. Circuits Syst., May 2012, pp. 121124
- [5] P. K. Meher and S. Y. Park, "Low adaptation-delay LMS adaptive filter part-I: Introducing a novel multiplication cell", in Proc. IEEE Int. Midwest Symp. Circuits Syst. Aug. 2011, pp. 14.
- [6] P. K. Meher and S. Y. Park, "Low adaptation-delay LMS adaptive filter part-II: An optimized architecture", in Proc. IEEE Int. Midwest Symp. Circuits Syst. Aug. 2011, pp. 14.
- [7] P. K. Meher and M. Maheshwari, "A high-speed FIR adaptive filter architecture using a modified delayed LMS algorithm", in Proc. IEEE Int. Symp. Circuits Syst., May 2011, pp. 121124.
- [8] L.-K. Ting, R. Woods and C. F. N. Cowan, "Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers", IEEE Trans. Very Large Scale Integr. (VLSI) Syst, vol. 13, no. 1, pp. 8699, Jan. 2005.
- [9] R. Rocher, D. Menard, O. Sentieys and P. Scalart, "Accuracy evaluation of fixed point LMS algorithm", in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process, May 2004, pp. 237240.
- [10] S. Haykin and B. Widrow, Least-Mean-Square Adaptive Filters Hoboken, NJ, USA: Wiley, 2003.
- [11] L. D. Van and W. S. Feng, "An efficient systolic architecture for the DLMS adaptive filter and its applications", IEEE Trans. Circuits Syst. II, Analog Digital Signal Process, vol. 48, no. 4, pp. 359366, Apr. 2001.
- [12] S. Ramanathan and V. Visvanathan, "A systolic architecture for LMS adaptive filtering with minimal adaptation delay", in Proc. Very Large Scale Integr. (VLSI) Design, Jan. 1996.
- [13] 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.