International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 26 Issue 4 – JUNE 2019.

Design and Implementation of High Speed LMS Adaptive Filtres Using Approximative Multipliers

[1]Mrs.Kalaiarasi.D ,[1]Mrs.S.Saranya, [2]D.Sunil,[3] K.D.Praveen Kumar ,[4] M.V.N.Sai Saran
[1]Assistant Professor, [2,3,4] U.G. Scholar, Panimalar Institute of Technology,Chennai.

ABSTRACT

Adaptive filters based on least mean square (LMS) algorithm constitute a standard in many DSP applications. The LMS algorithm, being an approximation of the wiener filter, is inherently imprecise, and constitutes a fertile ground to employ approximate hardware techniques with the additional challenge related to the presence of a feedback path for coefficients update. In this project, approximate LMS adaptive filters are explored for the first time, by employing approximate multipliers. A system identification scenario is adopted to assess the algorithm behavior. The analysis reveals that the choice of the approximate multiplier topology should be carefully examined; otherwise the stability and convergence performance of the algorithm can be compromised. The propose a novel approximate multiplier able to reduce the power dissipation in adaptive LMS filters with tolerable convergence error degradation.

KEYWORDS - Approximate computing; approximate multipliers; LMS; adaptive filters; Xilinx ISE.

1.INTRODUCTION:

Approximate computing enhance digital circuits performance by accepting erroneous computation [1]. In error resilient applications the error deriving from approximations result in tolerable quality loss.Example of error resilient applications are multimedia processing, in which, due to the perceptual limitation of human sensors, small errors can be tolerated [2]. Moreover, applications requiring an acceptable range of results in place of a signal "perfect" result, such as recognitions, mining and synthesis applications, can mitigate the effect of computational errors [3].

Adaptive filtering based on the Widrow-Hoff least mean square (LMS) algorithm [4], represents a standard in DSP applications for channel adaptive equalization, system identification, adaptive noise cancellation, etc. The LMS algorithm exhibits numerical stability satisfactory convergence error and computational simplicity [5].An LMS adaptive filter is usually composed by a FIR filter (due to its inherent stability) whose coefficients are updated by the LMS algorithm (fig.1)

Many papers on the focused on the reduction of complexity of the LMS algorithm, which is mainly due to the multipliers in the FIR filter and in the coefficient update circuitry (fig.1). In [6] a CORDIC version of the LMS algorithm is proposed, allowing to replace about fifty percent of multipliers with pipelined CORDIC units. The usage of distributed arithmetic (bit-serial operations and LUTs) is examine in [7]-[9] to reduce area occupation and power dissipation. A critical part analysis of the LMS algorithm is presented in [10]. Here the authors observed that, for the most practical cases, no-pipelining of the LMS algorithm is required, while, when high sampling rates are required (e.g. radar applications), delay-LMS (DLMS) can be adopted. In [11] the multiplication operations are realized has shit-and-add operations by adopting the SPT format, in the context of the sign-LMS family[12].

The LMS is a recursive algorithm aimed to minimize the mean-square-error (MSE) between the FIR filter output and a desired signal. The minimization requires MSE gradient computation, which, in the LMS algorithm is computed in an approximated way, causing the so called gradient noise [5]. To the best of authors knowledge, this is the first time that approximate circuits, like [13]-[14], are investigated in the context of adaptive LMS filtering.

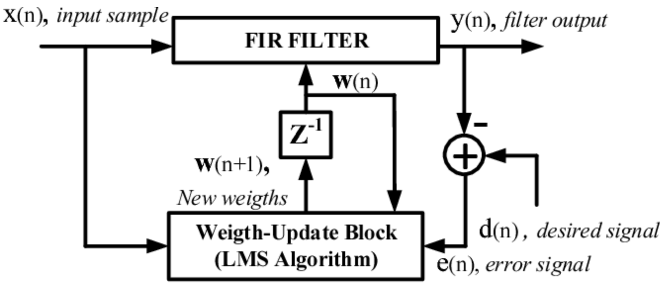


Fig. 1 Adaptive LMS filter.

In this paper we investigated the error-performance trade-off of approximate adaptive LMS filters, employing the multipliers proposed in [15]-[16] in a system identification applications. The analysis revels that the choice of the approximate multiplier topology must be carefully examined, otherwise, due to the presence of the feedback path, the stability and convergence performance of the algorithm can be compromised. To this purpose proposed a variation of the multiplier in [16], to optimize the error-performance trade-off. The proposed circuits are implemented in TSMC40nm technology, showing that adaptive LMS filter based on the proposed multiplier allow 29 % reduction of power dissipation with tolerable convergence error degradation.

2. WIDROW-HOFF (LMS) ALGORITHM:

Adaptive filtering based on the widrow -Hoff least-mean square (LMS) algorithm represents a standard in DSP applications for channel adaptive equalization, system identification, adaptive noise cancellation, etc. it is comprised of a finite-impulse-response (FIR) filter. The weights of FIR filter are updated using least mean square (LMS) algorithm due to its simplicity and satisfactory convergence. Generally , these filters consist of several multiply-and-accumulate

(MAC) unit depending upon tap-size

The LMS is a recursive algorithm aimed to minimize the mean-square-error (MSE) between the FIR filter output and a desired signal. The minimization requires MSE gradient computation, which, in the LMS algorithm, is computed in an approximated way, causing the so-called gradient noise. Therefore, the LMS algorithm is characterized by an inherent grade of noise and constitutes a fertile ground to employ approximate hardware circuits.

3. APPROXIMATE LMS ADAPTIVE FILTERS:

The paper investigates the error-performance trade-off of approximate adaptive LMS filters, employing the multipliers proposed in a system identification application. The analysis reveals that the choice of the approximated multiplier topology must be carefully examined, otherwise, due to the presence of the feedback path, the stability and convergence performance of the algorithm can be compromised. To this purpose the propose a variation of the multiplier, to optimize the error-performance trade-off. The proposed circuits are implemented, and showing that adaptive LMS filters based on the proposed multiplier allow reduction of power dissipation with tolerable convergence error degradation.

Block diagram:

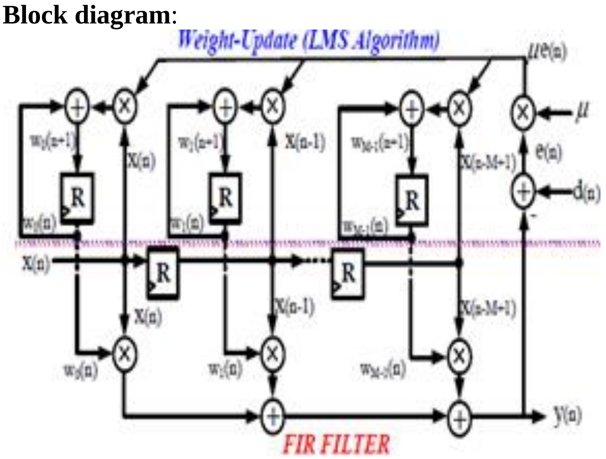
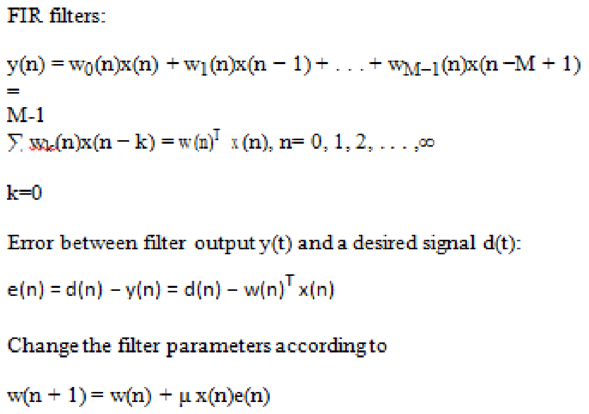


Fig.3.1. Adaptive LMS filter-hardware overview

4.MODULE EXPLANATION

4.1 LMS Algorithm:

Itis a stochastic gradient descent method in that the filter is adapted based on the error at the current time. It was invented in year 1960 by Stanford university professor BenardWidrow and his first ph.d.student, Ted Hoff. LMS is linear adoptive filter algorithm and it is consisted of filtering processes and adopting process.



FIR filters:

$y(n) = w_0(n)x(n) + w_1(n)x(n-1) + \ldots + w_{M-1}(n)x(n-M+1)$

=

M-1

$\sum w_k(n)x(n-k) = w(n)^T\ x(n)$, n= 0, 1, 2, . . . ,∞

k=0

Error between filter output y(t) and a desired signal d(t):

$e(n) = d(n) - y(n) = d(n) - w(n)^T x(n)$

Change the filter parameters according to

$w(n+1) = w(n) + \mu\, x(n)e(n)$

LMS algorithm uses a rough gradient approximation, and seeks and wished weight vector this process is used to find the weight vector for training the ALC (Adaline). The learning rules can be incorporate to the same device that therefore can be auto updated as there are presented at the wised inputs and outputs. The weight vector values are changed very combination input-output is processed. This goes on until the ALC gives the correct outputs. This is a truly training process since there is not necessary to clearly calculate the weight vector value

4.2 BRIEF REVIEW OF THE LMS ALGORITHM:

Note that, different from non-adaptive FIR filters, in (1) the weights are a function of the time instant *n*. In adaptive LMS filters (Fig. 1) the difference between the actual filter output

y (*n*)and the desired signal *d* (*n*) defines the error signal

e(*n*):

$e(n) = d(n) - y(n)$

The error $e(n)$ is used by the LMS algorithm to determine the updated version of the filter tap weights $w_k(n)$, as follows:

$$w_k(n+1) = w_k(n) - \mu \cdot e(n) \cdot x(n-k)$$

Where μ is the step-size parameter, determining a trade-off between the convergence speed and the convergence error of the algorithm. Please note that the term $\mu \cdot e(n) \cdot x(n-k)$ is an approximation [5] of the gradient of the MSE cost function.

Therefore LMS algorithm minimizes the MSC cost function in an approximated sense, due to the approximation on the gradient (gradient noise) (note that this gradient is computed exactly in the winner filter [5], which is, however, very hard to be implemented in hardware).Fig.2 shows the adaptive LSM filter described by (1)-(3) from a circuital perceptive. Note that the direct form FIR filter is employed in this paper. Since it exhibits faster convergence and lesser registers number then it transpose-form implementation, [10]. Moreover, the step-size parameters is chosen has a power of two, to implement the corresponding multiplication operation with a hardwired shift.

4.3 Adaptive filters:

The adaptive algorithm uses the value of criterion of performance, the measurements of the input and desired signals so as to modify the parameters of the filter to improve its performanceAdaptive filters are designed for a specific type of input signal(speech, binary data, etc.), for specific types of interference (adaptive white noise, sinusoidal signals, echoes of the input signals, etc.), and for specific types of signal transmission paths (e.g., linear time-invariant or time varying).

Desired

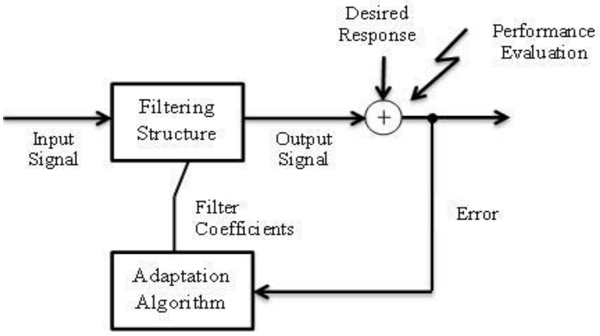


Fig.4.3. Adaptive filters

After choosing the filtering structure, criterion of performance and the adaptive algorithm, the only unknowns, [17]. If the characteristics of the relevant signals are constant , the goals of the adaptive filters is to find the parameters that give the best performance and then to stop the adjustment. However, when the characteristics of the relevant signals change with time, the adaptive filter should first find and then continuously readjust its parameters track this changes. In the current implementation of adaptive filter, the criterion of performance is mean square value of the estimation error which is used to optimize the filter design[18]. The Mean Square Error (MSE) criterion at second order depends on the cost function described by the error performance surface

4.4Approximate Multiplier:

Multipliers are most important parts in signal processing applications or other computationally drastic applications. Therefore, multiplier designs are mainly focused on high-speed, low area and low power. These parameters are achieved by approximate multipliers. Generally, approximate computing has a significant attention as a rising strategy to decrease power consumption of error tolerant applications like image processing. Approximate computing has been advocated as a new approach to saving area, as well as increasing performance at a limited loss in accuracy. The partial product matrix (PPM) of approximate multipliers consist of approximate 2x2 multiplier block, which takes four partial products, belonging to three adjacent columns of the PPM, and computes an approximated sum, producing three output bits (in place of four). The most significant and the least significant partial products of each block are provided unchanged as output, while the remaining output bit is obtained by OR-ing the two middle partial products (in this way, an OR gate is used in place of a half-adder). The 2x2 block is replicated along the whole PPM to implement the approximate multiplier.Signed multipliers are often needed in practical DSP applications (such as LMS filtering), therefore in this section we review and the approximate unsigned multipliers to the signed case.

4.4.1 kulkarani:

The kulkarni approximate multiplier is shown.

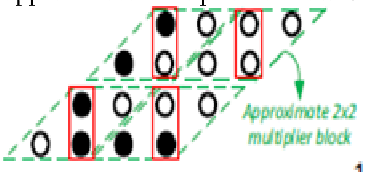


Fig.4.4.1.kulkarani

1.It is based on approximate 2*2 multiplier block, which takes four partial products, belonging to three adjacent columns of the PPM, and computes an approximated sum, producing three outputs bits (in place of four).

2.the most significant and the least significant partial products of each block are provided unchanged as output, while remaining output bit is obtained by OR-ing the middle partial products (in this way, an OR gate is used in place of a half adder).

3.the 2*2 block is replicated along the whole PPM to implement the approximate multiplier.

4.the error-performance trade-off can be tuned (at design time)by using exact 2*2 blocks in the most-significant columns of the multiplier PPM.

4.4.2 approxmul2:

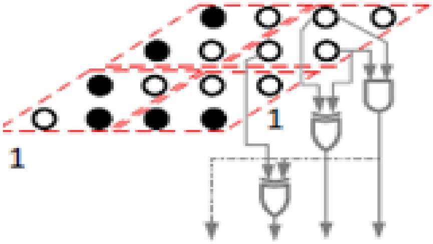


Fig.4.4.2.approxmul2

1. It is composed by the replication of 2*2 approximate blocks (like the kulkarni multiplier), but here each blocks is more complex than kulkarni one.

2. This reduces the maximum error but increases the error probability and worsens the electrical performance (it employs two XOR and one AND gates).

4.4.3 date17:

The Fig shows the DATE17 in which the partial products are compressed by using OR gates in place of half-adders to reduce the PPM height.

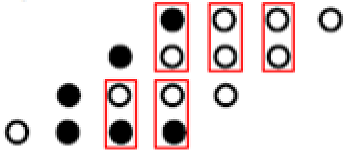


Fig.4.4.3.date17

1.Note that in the original version of the paper two, three and four inputs OR gates are used as compressors, producing a single bit. In the following we will consider the case of 2-inputs OR gate only, since, in the remaining configurations, the approximation error is too large for the proposed application.

4.4.4 prime17:

In this fig the approximate multiplier is shown.

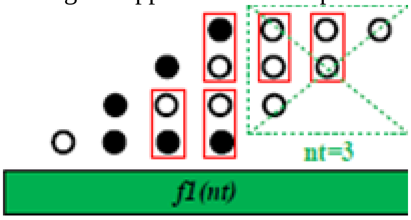


Fig.4.4.4.prime17

1 .Here the compression scheme is similar to the previous one, but, to improve the electrical performance, some least significant PPM columns are truncated (at design time).

2. The resulting mean error is compensated employing a constant term f1 (nt) (where nt is the number of truncated columns).

In this fig the proposed multiplier is shown.

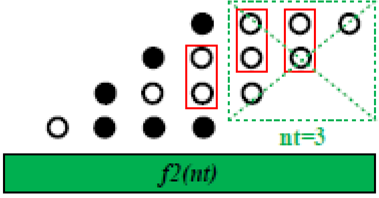


Fig. proposed multiplier

1. The proposed multiplier, in which the approximate compression of NAND partial product is avoided.

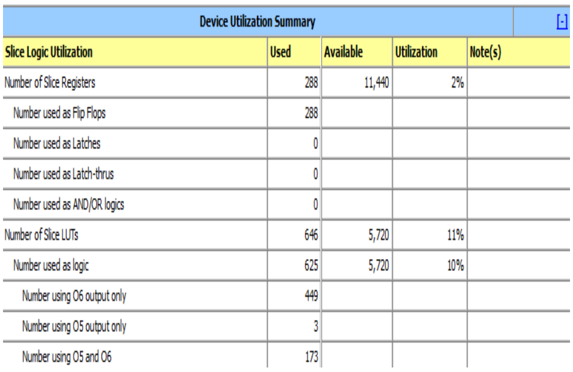
2.The error probability is three times higher when the compression involves NAND partial products.

3.More over, to reduce the error accumulation due to the feedback of the LMS filter, the approximate compression only involves the n least significant columns. The mean error is compensated with a suitable constant f2 (nt).

5. OUTPUT WAVEFORMS:

5.1 ADAPTIVE FILTERS:

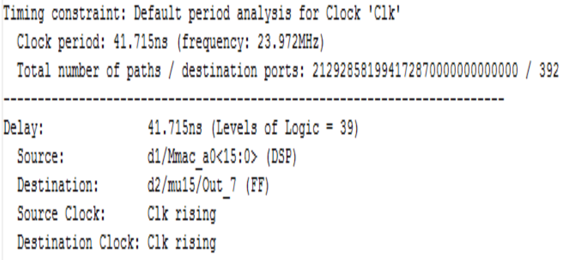
DESIGN SUMMARY:



Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	288	11,440	2%	
Number used as Flip Flops	288			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	646	5,720	11%	
Number used as logic	625	5,720	10%	
Number using O6 output only	449			
Number using O5 output only	3			
Number using O5 and O6	173			

Fig.5.1.1.design summary table

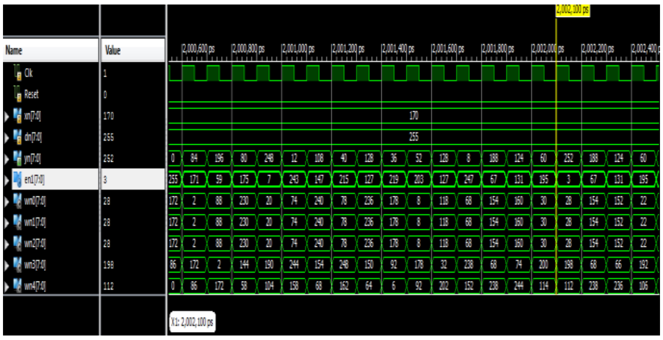
TIMING ANALYSIS:



```
Timing constraint: Default period analysis for Clock 'Clk'
  Clock period: 41.715ns (frequency: 23.972MHz)
  Total number of paths / destination ports: 21292858199417287000000000000 / 392
-------------------------------------------------------------------------
Delay:               41.715ns (Levels of Logic = 39)
  Source:            d1/Mmac_a0<15:0> (DSP)
  Destination:       d2/mu15/Out_7 (FF)
  Source Clock:      Clk rising
  Destination Clock: Clk rising
```

Fig.5.1.2.timing report analysis

STIMULATION:

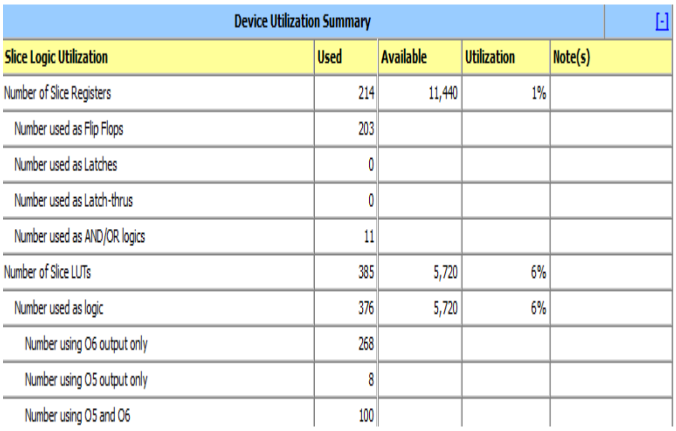


Name	Value
Clk	1
Reset	0
xn[7:0]	170
dn[7:0]	255
yn[7:0]	252
en1[7:0]	3
wn0[7:0]	28
wn1[7:0]	28
wn2[7:0]	28
wn3[7:0]	198
wn4[7:0]	112

Fig.5.1.3. stimulation output

5.2 APPROXIMATE MULTIPLIERS:

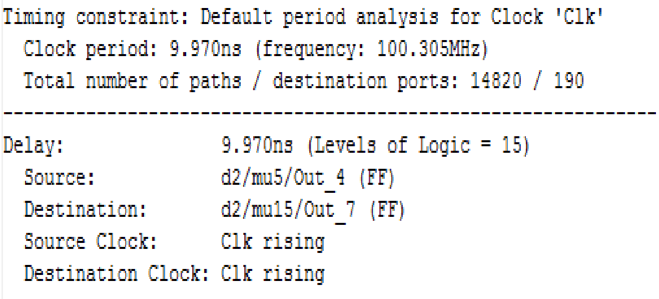
DESIGN SUMMARY:



Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	214	11,440	1%	
Number used as Flip Flops	203			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	11			
Number of Slice LUTs	385	5,720	6%	
Number used as logic	376	5,720	6%	
Number using O6 output only	268			
Number using O5 output only	8			
Number using O5 and O6	100			

Fig.5.2.1.design summary table

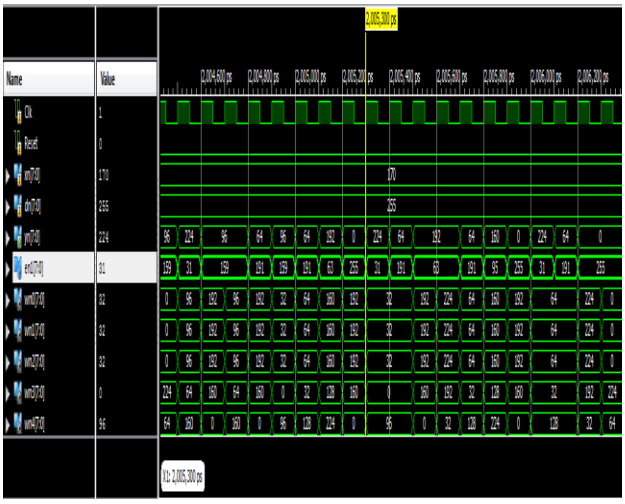
TIMING ANALYSIS:



```
Timing constraint: Default period analysis for Clock 'Clk'
  Clock period: 9.970ns (frequency: 100.305MHz)
  Total number of paths / destination ports: 14820 / 190
-------------------------------------------------------------------------
Delay:               9.970ns (Levels of Logic = 15)
  Source:            d2/mu5/Out_4 (FF)
  Destination:       d2/mu15/Out_7 (FF)
  Source Clock:      Clk rising
  Destination Clock: Clk rising
```

Fig.5.2.2. timing report analysis

STIMULATION OUTPUT:



Name	Value
Clk	1
Reset	0
xn[7:0]	170
dn[7:0]	255
yn[7:0]	224
en1[7:0]	31
wn0[7:0]	32
wn1[7:0]	32
wn2[7:0]	32
wn3[7:0]	0
wn4[7:0]	96

Fig.5.2.3. stimulation output

CONCLUSION:

Adaptive LMS filters using approximate multipliers in the FIR filter section have been investigated for the first time in this paper. The analysis reveals that, due to the feedback loop for the coefficient update, non-aggressive approximate multipliers must be employed. A novel approximate multiplier is proposed, employing optimized partial product approximate compression, least significant columns truncation and mean error compensation. Implementation results reveal that adaptive LMS filters based on the proposed multiplier exhibit very-good quality-power trade-off, allowing power reduction with good convergence properties

REFERENCES:

[1] J. Han, M. Orshansky, "Approximate computing: an emerging paradigm for energy-efficient design", proc. 18th IEEE Europ. Test Symp., pp. 1-6, May 2013.

[2] V. K. Chippa, S. T. Chakradhar, K. Roy and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, 2013, pp. 1-9.

[3] Q. Xu, T. Mytkowicz and N. S. Kim, "Approximate Computing: A Survey," in IEEE Design & Test, vol. 33, no. 1, pp. 8-22, Feb. 2016.

[4] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood
Cliffs, NJ, USA: Prentice-Hall, 1985.

[5] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, 2002.

[6] M. Chakraborty, A. S. Dhar and Moon Ho Lee, "A trigonometric formulation of the LMS algorithm for realization on pipelined CORDIC," in *IEEE Transactions on Circuits and Systems II: ExpressBriefs*, vol. 52, no. 9, pp. 530-534, Sept. 2005.

[7] D. J. Allred, HeejongYoo, V. Krishnan, W. Huang and D. V. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 7, pp. 1327-1337, July 2005.

[8] M. T. Khan and S. R. Ahamed, "A New High Performance VLSI Architecture for LMS Adaptive Filter Using Distributed Arithmetic," *2017 IEEE Computer Society Annual Symposium on VLSI(ISVLSI)*, Bochum, 2017, pp. 219-224.

[9] M. T. Khan, S. R. Ahamed and F. Brewer, "Low Complexity and Critical Path Based VLSI Architecture for LMS Adaptive Filter Using Distributed Arithmetic," *2017 30th International Conference on VLSIDesign and 2017 16th International Conference on Embedded Systems (VLSID)*, Hyderabad, 2017, pp. 127-132.

[10] P. K. Meher and S. Y. Park, "Critical-Path Analysis and Low-Complexity Implementation of the LMS Adaptive Algorithm," in *IEEETransactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 3,

pp. 778-788, March 2014.

[11] S. Choudhary, P. Mukherjee, M. Chakraborty and S. S. Rath, "A SPT Treatment to the Realization of the Sign-LMS Based Adaptive Filters," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 9, pp. 2025-2033, Sept. 2012.

[12] B. Farhang-Boroujeny, *Adaptive Filters: Theory and Applications*, John Wiley & Sons, 1998.

[13] D. Esposito, D. De Caro, A.G.M. Strollo, “Variable Latency Speculative Parallel Prefix Adders for Unsigned and Signed Operands”, IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 63, n. 8,

pp. 1200-1209, Aug. 2016.

[14] D. Esposito, G. Castellano, D. De Caro, E. Napoli, N. Petra and A.G.M. Strollo, "Approximate adder with output correction for error tolerant applications and Gaussian distributed inputs," 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, 2016, pp. 1970-1973.

[15] C. Liu, J. Han, F. Lombardi, “An Analytical Framework for Evaluating the Error Characteristics of Approximate Adders”, IEEE Transactions on Computers, vol. 64, no. 5, pp. 1268-1281, May 2015.

[16] E. J. King, E. E. Swartzlander, Jr., “Data dependent truncated scheme for parallel multiplication”, Proc. of 31th Asilomar Conf. on signals, circuits and systems, pp. 1178-1182, 1997.

[17] DimitrisG.Manolakis, VinayK.Ingle, Stephen M.Kogon, Statistical and Adaptive Signal Processing, Artech House, MA, 2005.

[18] M.Stojanovic, J.Proakis and J.Catipovic, "Performance of High-Rate Adaptive Equalization on a Shallow Water Acoustic Channel," J. Acoust. Soc. Am., vol.100 (4), Pt.1, pp.2213-2219, October 1996.