International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

Power Efficient LMS Adaptive Filters using Approximate Compressors

Sivakami Ramakrishnan
VLSI Design (M.E.)
Electronics and Communication Engineering
Jeppiaar Engineering College
Chennai, India

Mrs. Nanammal
Assistant Professor
Electronics and Communication Engineering
Jeppiaar Engineering College
Chennai, India

***Abstract*:- Adaptive filters based on least-mean-square algorithm constitute a standard in many Digital Signal Processing applications. The LMS algorithm, is a part of the Wiener filter, that constitutes a fertile ground to employ approximate hardware techniques with the additional challenge related to the presence of a feedback path for coefficients update. Here the, approximate LMS adaptive filters are explored for the first time, by employing approximate multipliers along with the approximate compressors. A system identification scenario is adopted to assess the algorithmic behavior. The choice of the approximate multiplier should be carefully examined; because the stability and convergence performance of the algorithm can be compromised. The novel approximate multiplier able to reduce the power dissipation in adaptive LMS filters with tolerable convergence error degradation**

Keywords—DSP, LMS, approximation.

I. INTRODUCTION

The digital circuits are considered in such a way that the order is not met in the terms of functionality and reduction is required in terms of area; delay and energy are called approximate circuits [1]. Approximate computation is an emerging development in the digital design [1]. This approach has become more and more significant for the embedded and mobile systems, characterized by the severe energy and speed constraint [4]. An LMS adaptive filter is usually composed by a FIR filter (due to its inherent stability) whose coefficients are updated by the LMS algorithm. 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, be computed in an approximated way, causing the so-called gradient noise [13]. A multiplier involves a few basic blocks: They are (1) partial product's generation, (2) partial products reduction and (3) carry-propagate addition. The Approximations can be introduced in any one of these blocks. In this paper, novel approximate multiplier is proposed by a simple, fast approximate adder. This lately designed adder will be able to process the data in the parallel by removing the carry propagation chain (introducing an error). It has a critical path delay that is still shorter than the conventional one-bit full adder [2]. In this proposed approximate multiplier, a simple approximate adder is used for partial product accumulation, and the error signals are used to compensate the error for obtaining a better accuracy. Compared to other multipliers, the proposed multiplier has an appreciably shorter critical path and reduced circuit complexity [2].

The LMS algorithm is a recursive algorithm aimed to reduce the mean-square-error (MSE) between the FIR filter output and a desired signal. The minimization requires MSE gradient computation, in which the LMS algorithm is computed in an approximated way, and hence called as gradient noise [13]. Therefore, the LMS algorithm is characterized by an inherent grade of noise and constitutes a fertile ground to employ any hardware circuits. This is the first time that approximate circuits, is used in the context of adaptive LMS filtering [3]. We intend the use of approximate multipliers in the FIR section of the adaptive filter, since (i) multipliers consist of one of the most energy-hungry digital blocks and (ii) multiplication is the most used operation in adaptive LMS filters. In this paper we examine that the error-performance of approximate adaptive LMS filters, employing the multipliers proposed in a system identification process [14]. The analysis reveals that the selection of the approximate multiplier topology should be carefully examined, or else, due to the presence of the feedback path, the stability and the convergence performance of the algorithm can be compromised [4]. The proposed circuits are implemented showing that adaptive LMS filters based on the proposed multiplier allows reduction of power dissipation with acceptable convergence error degradation.

II. LMS ALGORITHM

First, Least mean squares algorithms (LMS Algorithm) are a class of adaptive filter that is used to mimic a desired filter by finding the filter coefficients that relay to produce the least mean square of the error signal. It is a stochastic gradient method in which the filter is only adapted based on the error at the present time. Consider the LMS ALGORITHM as shown in the Figure – 1.

Let us consider an M-taps FIR filter, with tap weights $w_k(n)$,$k \in \{0,1,...,M-1\}$ and tap inputs $x(n) = [x(n), x(n-1),..., x(n-M+1)]$. The output at each time instant n can be written as:

$$Y(n) = \sum_{k=0}^{M-1} w_k(n).x(n-k) \qquad (1)$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

In adaptive LMS filters 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). (2)

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

wk(n +1) = wk(n) −μ ·e(n)·x(n − k). (3)

μ - step-size parameter, which relates a trade-off between the convergence speed and convergence error of the algorithm. The LMS algorithm reduces the MSE cost function approximately; this is due to the approximation on the gradient noise. Note that the term μ· e(n)·x(n-k) is an approximation of the gradient of the MSE cost function J(n) = e[$|n^2|$], where E is the statistical expectation operation[2].

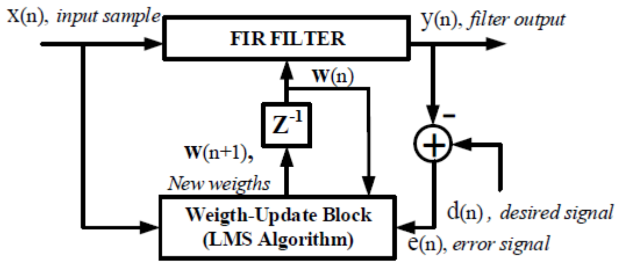

Figure – 1. Adaptive Filter

III. PROPOSED SYSTEM

In this existing system, we have used approximate multipliers for the multiplication operation and approximate compressors for the partial product addition. Thus it reduces the area and complexity in this architecture. The approximate multipliers and approximate compressor are as follows;

A. Approximate Multipliers

Approximate multipliers are frequently used in real time DSP applications such as LMS filtering. They are designed in such a way that the condition such as in terms of energy, delay or area are met and they do not meet in terms of functionality, and these type of circuits are called as approximate circuits. Approximate computing is a computation that provides all the possible result that are available and the required sufficient result will also be presented. For example, search engine provides all the result that are related with the search, from which we can select the required result. They are perfectly suitable for some of the application such as multimedia applications (e.g. text, graphics, images, sound/audio, animation and/or video), data mining applications (e.g. Financial Data Analysis, Industry, Telecommunication Industry, Biological Data Analysis**)** [5]**.** Approximation techniques that are used in the multipliers focuses on the accumulation of the partial products, which is decisive when it comes to power consumption.

B. Approximate Compressors

Approximate compressor are uses to compress the number of gate used in this architecture, hence the area as well as the time for computation is reduced

Let us consider to unsigned n bits
X = $x_{n-1}2^{n-1}$+……..+x_0, and
Y = $y_{n-1}2^{n-1}$+……....+y_0

Z = X*Y = $p2_{n-1}2^{2n-1}$+………+p_0 ……. (1)

where Z is the product of two inputs X and Y as shown in equation (1).
As shown in figure. 2. Let us consider the partial products belonging to the j-th column of the Partial product multiplier are given as:
$p_0 = x_{j-1}y_0, p_1 = x_{j-2}y_1 \dots\dots\dots, p_{j-1} = x_0 y_{j-1}$.
The sum of this partial product is given as S and the value of S ranges from 0 to j. It can be given as S = $\sum\{p_0, p_1, \dots\dots, p_{j-1}\}$.The compressor have 3 inputs and two outputs they are sum and carry.

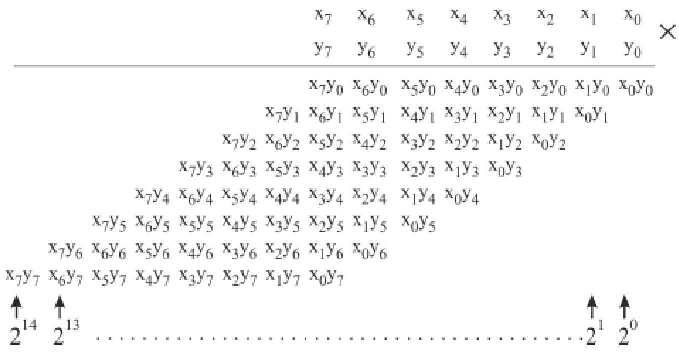

Figure – 2. Partial Product of 8x8 multipler

These approximate compressors have j inputs $p_0, p_1, \dots\dots, p_{j-1}$ and the output extends till (j-1) by using a novel approximation which is aimed to reduce the probability of error. Here the compressor inputs are partial products that arrive from the PPM column. Moreover, the input bits xi and yj are uniform and are independent of each other. The probability of the partial product is expressed as: P (pi) = 1/4.

1) Approximate 2/1 Compressor

Consider the summing of two partial products in the same column. As we have seen in [6], [7], [9] the sum is given as:

SUM = $\sum\{p_0, p_1\} = \sum\{p_0 p_1, p_0 + p_1\}$ …..(2)

The sum of the two partial products (2), is given by the logic of AND and the logic of OR. The probability (3) of the two terms is given as:

P($p_0 p_1$) = 1\16 and P($p_0 + p_1$) = 7\16 ……(3)

The approximate arithmetic sum of two partial products can be approximated as:

$S_{APPROX} = \sum p_0 + p_1$ …… (4)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

According to the behavior of the approximate 2/1 compressor the output is indicated as w1. While E_{RR} (error) indicates the difference between actual value and approximate compressor value is given by the equation(5):

$$E_{RR} = S - S_{APPROX} \qquad \ldots\ldots(5)$$

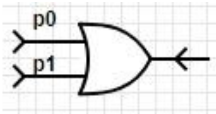

Figure – 3. Schematic of 2/1 compressor

Table – 1
Approximate 2/1 Compressors

p_1	p_0	w_1	S	S_{APPROX}	E_{RR}
0	0	0	0	0	0
0	1	1	1	1	0
1	0	1	1	1	0
1	1	1	2	**1**	1

The error probability P_E and the mean error E_{mean} is given by the following equation:

$$P_E = \sum_i P\, E_{RRi} \qquad \ldots\ldots(6)$$

$$E_{mean} = \sum_i P\, E_{RRi}(E_{RRi}) \qquad \ldots..(7)$$

2) Approximate 3/2 Compressor

Consider the summing of three partial products in the same column. The sum is given as:

$$\text{SUM} = \sum\{p_0, p_1, p_2\} = \sum\{p_0 p_1, p_0 + p_1, p_2\} \qquad \ldots..(8)$$

The sum of the three partial products (8), is given by the logic of AND and the logic of OR.

$$\text{SUM} = \sum\{p_0 p_1 p_2, p_0 p_1 + p_2, p_0 + p_1\} \qquad \ldots..(9)$$

The probability (9) of the three terms is given as:

$$P(p_0 p_1 p_2) = 1\backslash 64 \qquad \ldots\ldots..(10)$$

The approximate arithmetic sum of three partial products can be approximated as:

$$S_{APPROX} = \sum p_0 p_1 + p_2, p_0 + p_1 \qquad \ldots\ldots(11)$$

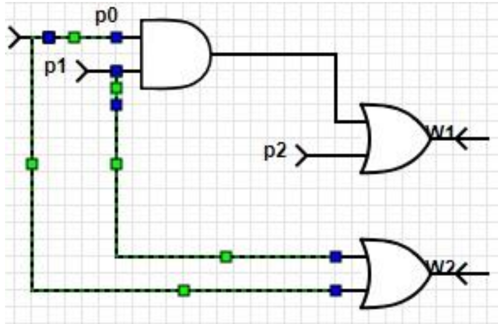

Figure – 4. Schematic of 3/2 compressor

Table – 2
Approximate 3/2 Compressors

p_2	p_1	p_0	w_2	w_1	S	S_{APPROX}	E_{RR}
0	0	0	0	0	0	0	0
0	0	1	0	1	1	1	0
0	1	0	0	1	1	1	0
0	1	1	1	1	2	2	0
1	0	0	1	0	1	1	0
1	0	1	1	1	2	2	0
1	1	0	1	1	2	2	0
1	1	1	1	1	3	**2**	1

The error probability P_E is given in the closed form. Let us consider two sub-compressors such as A and B are exploited, the error probability is given as:

$$P_E = P_E(A) + P_E(B) - P_E(A)P_E(B)$$

Now let us consider three sub-compressors such as A,B and C, the error probability is given as:

$$P_E = P_E(A) + P_E(B) + P_E(C) - P_E(A)P_E(B) - P_E(B)P_E(C) - P_E(C)P_E(A) + P_E(A)P_E(B)P_E(C)$$

This equation can also be extended to wide-ranging case and it is called as the Inclusion-Exclusion principle [12]. This shows that i) the probability error and the mean error tends to increase with the compressor size and ii) the approximate compressors having an odd number of input performs better when compared to the compressors having an even number of input bits.

IV. RESULT AND DISCUSSION

The adaptive filter using approximate multiplier along with the approximate compressors have been exploited to design 8 × 8, 12 × 12, 16 × 16, and 20 × 20, binary multipliers. The prediction reveals that, because of the presence of the feedback loop in the filter, that is used to update the weight coefficient approximate multipliers must be introduced. This proposed system is compared adjacent to the exact multipliers and with the approximate multipliers that are proposed in [11], in [10] and in [8] using 2, 3 or 4-input OR gates as approximate compressor.

A. Power Quality Trade Off And Electrical Performance:

The waveform of the existing (Figure – 5a) and the proposed (Figure – 5b) system is given as follows:

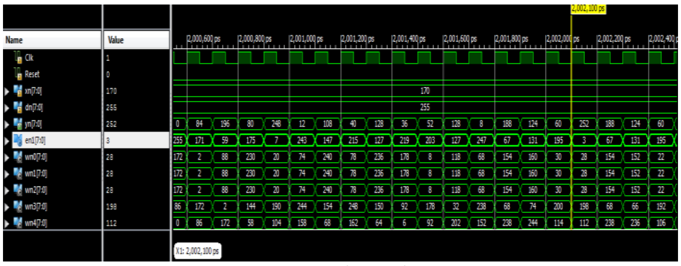

Figure – 5a. Waveform of existing system

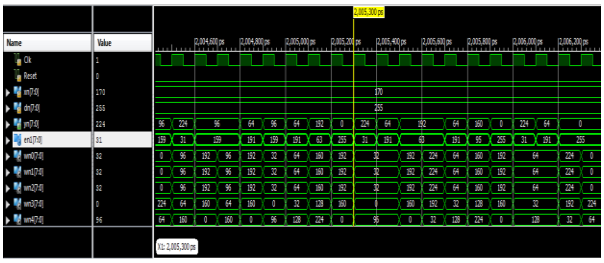

Figure – 5b Waveform of proposed system

This adaptive LMS filters is checked with the Verilog HDL and standard cell. Here the Power dissipation is calculated from the SDF and TCF – which is based on post-synthesis simulations. According to approximations, the filters based on the proposed multiplier, shows improvement up to 16-18% in area. The below table shows the improvement from the existing (Figure – 6a) to that of the proposed one (Figure – 6b).

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			

Figure – 6a. Design summary of existing

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			

Figure – 6b. Design summary of proposed

Table – 3. Comparison of area between existing and proposed

MULTIPLIERS	AREA IN LUT's
EXISTING	646
PROPOSED	385

Thus the comparison of area between the existing one and the proposed one shows that reduced number of LUT's have been used for the proposed approximate multipliers. Thus using the proposed approximate multiplier with (nt=0) has the lowest error of 10%.

B. Error Performance

The errors that are considered in the following approximate multipliers are:
- The probability of Error Rate (Pe)
- The mean square errors given by the multiplier (ERMS).

Here introducing the Number of Effective bits, which can also be represented as - NoEB, of a approximate multipliers can also be defined as:
NoEB= 2n − log2 (1 + ERMS) (for a n×n multipliers). Here we use NoEB because they give immediate and accurate or sometimes approximate number of output bits that are free from error.

The error can be determined by the difference between the approximate output to that of the exact output values. They are given as:
$E = Y_{Exact} - Y_{Approx}$. Where Y_{Exact} represents the output computed by the exact multiplier and Y_{Approx} represents the output computed by the approximate multiplier. With the help of this error metrics we can average four sizes (8 × 8, 12 × 12, 16 × 16, and 20 × 20).

V. CONCLUSION

This paper represents the use of approximate multipliers and approximate compressors that overcomes other type of multipliers in terms of complexity in the circuit and error performance. Thus it can also can be given as approximate multiplier is introduced, by employing approximate compression, and hence LMS columns are being truncated. This implementation expose that adaptive LMS filters based on multiplier and compressor provides a high quality power trade-off, and also helps in the reduction of power. Along with the effect of the introduction of approximate multipliers in the Adaptive LMS filter is examined, with reference to the performance of the algorithm and the electrical performances.

REFERENCES

[1] J. Han and M. Orshansky, “Approximate computing: An emerging paradigm for energy-efficient design,” in Proc. 18th IEEE Eur. TestSymp., May 2013, pp. 1–6.
[2] P. Kulkarni, P. Gupta, and M. Ercegovac, “Trading accuracy for power with an underdesigned multiplier architecture,” in Proc. 24th Int. Conf.VLSI Des., Jan. 2011, pp. 346–351.
[3] L. Dadda, “Some schemes for parallel multipliers,” Alta Freq., vol. 34, pp. 349–356, Mar. 1965.
[4] S. Veeramachaneni, K. M. Krishna, L. Avinash, S. R. Puppala, and M. B. Srinivas, “Novel architectures for high-speed and low-power 3-2, 4-2 and 5-2 compressors,” in Proc. 20th Int. Conf. VLSI Des. (VLSID), 2007, pp. 324–329.
[5] Q. Xu, T. Mytkowicz, and N. S. Kim, “Approximate computing: A survey,” IEEE Des. Test, vol. 33, no. 1, pp. 8–22, Feb. 2016.
[6] A. Cilardoet al., “High speed speculative multipliers based on speculative carry-save tree,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 61, no. 12, pp. 3426–3435, Dec. 2014.

[7] S. Venkatachalam and S.-B. Ko, “Design of power and area efficient approximate multipliers,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 5, pp. 1782–1786, May 2017.

[8] I. Qiqieh, R. Shafik, G. Tarawneh, D. Sokolov, and A. Yakovlev, “Energy-efficient approximate multiplier design using bit significance driven logic compression,” in Proc. Des., Automat. Test Eur. Conf.Exhib. (DATE), 2017, pp. 7–12.

[9] D. Esposito, A. G. M. Strollo, and M. Alioto, “Low-power approximate MAC unit,” in Proc. IEEE PRIME, Giardini Naxos, Italy, Jun. 2017, pp. 81–84.

[10] S. Rehman, W. El-Harouni, M. Shafique, A. Kumar, and J. Henkel, “Architectural-space exploration of approximate multipliers,” in Proc.IEEE/ACM Int. Conf. Comput.-Aided Des. (ICCAD), Austin, TX, USA, Nov. 2016, pp. 1–8.

[11] L.Dadda, “Some schemes for parallel multipliers,” Alta Freq., vol. 34, pp. 349–356, Mar. 1965.

[12] J. Riordan, An Introduction to Combinatorial Analysis. Hoboken, NJ, USA: Wiley, 1958.

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

[14] P. Kulkarni, P. Gupta and M. Ercegovac, "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," 2011 24th InternatioalConference on VLSI Design, Chennai, 2011, pp. 346-351.