ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 5, Issue 5, May 2018

Power and Area Efficient Approximate Multipliers for Image Processing Application

[1] Rakesh Huidrom, [2] Dr. R. Nagaraj Ramrao
[1] Department of Electronics & Communication, the Oxford College of Engineering, Bangalore, India
[2] Professor, Department of Electronics & Communication, the Oxford College of Engineering, Bangalore, India

Abstract- **Inexact or approximate computing can decrease the complexity of the design, thereby increasing its performance and power efficiency for error-tolerant applications. This brief deals with a new outline approach for inexact multipliers. The partial products results obtain for the multiplier are altered for producing varying probable terms. The logic complexity for inexact computing depends on the probability for the accumulation of altered partial products. The proposed inexact multiplier is utilized in two variations of 16-bit multipliers. They consume less power and area when compared to existing inexact multipliers. The performance of the proposed multipliers is estimated using image processing application, where it achieves the highest peak signal to noise ratio.**

***Index Terms*— inexact computing, error tolerant analysis, low power.**

I. INTRODUCTION

In applications like multimedia and digital signal processing which can tolerate error, exact computations are not always required. Approximate counterparts can be used to replace them. It brings the rise in the research on inexact computing for error-tolerant applications. Applications like Adders and multipliers form the key components in these. In digital signal processing applications, approximate full adders are proposed at transistor level and they are utilized [1]. In an accumulation of partial products, proposed full adders are used in multipliers. In fixed-width multiplier designs, truncation is widely employed to reduced hardware complexity of multipliers. To compensate for the quantization error introduced by the truncated part, a constant or variable correction term are added [2], [3]. In terms of power consumption, approximation techniques in multipliers focus are crucial on the accumulation of partial products. While forming partial products to reduced hardware complexity, broken array multiplier is implemented in the least significant bits of inputs are truncated. In partial product accumulation, the proposed multiplier saves few adder circuits [4]. In partial product reduction tree of four variants of 8 × 8 Dadda multiplier, two designs of approximate 4-2 compressors are presented and used [5]. For partial product accumulation of the multiplier, a new approximate adder is presented [10]. 26% of power is reduced is accomplished in 16-bit approximate multiplier as compare to exact multiplier [10]. Reduction of focus on the straightforward application of approximate adders and compressors to the partial products in the previous works on logic complexity. In this brief, the partial products are altered to introduce terms with different probabilities. Probability statistics of the altered partial products are analyzed, which is followed by systematic approximation. Simplified arithmetic units (half-adder, full-adder, and 4-2 compressor) are proposed for approximation. The arithmetic units are not only reduced in complexity, but care is also taken that error value is maintained low. While systemic approximation helps in achieving better accuracy, the reduced logic complexity of approximate arithmetic units consumes less power and area. The proposed multipliers outperform the existing multiplier designs in terms of area, power, and error, and achieves better peak signal to noise ratio (PSNR) values in image processing application. The rest of this brief is organized as follows. Section II details the proposed architecture. Section III provides extensive result analysis of the design of the proposed and enhancement approximate multipliers. The proposed multipliers are utilized in image processing application and results are provided in Section IV. Section V concludes this brief.

II. PROPOSED ARCHITECTURE

It comprises of following three steps in the implementation of the multiplier. They are as follows i) Generation of partial products, ii) Partial products reduction tree and iii) A vector merge addition to produce the final product from the sum and carry rows generated from the reduction tree. In this brief, an approximation is applied in reduction tree stage.

Consider two 8-bit unsigned input operands $\alpha = \sum_{m=0}^{7} a_m 2^m$ and $\square = \sum_{n=0}^{7} \square_n 2^n$.

$p_{m,n} = a_{m,n} + a_{n,m}$

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 5, Issue 5, May 2018

$g_{m,n} = a_{m,n} \cdot a_{n,m}$.

The probability of the altered partial product g_{mn} being one is 1/16, which is significantly lower than 1/4 of a_{mn}. The probability of altered partial product p_{mn} being one is 1/16 + 3/16 + 3/16 = 7/16, which is higher than g_{mn}. These factors are considered, while applying approximation to the altered partial product matrix.

A. Approximation of Altered Partial Products $g_{m,n}$

The accumulation of generate signals is done column wise. As each element has a probability of 1/16 of

TABLE I
PROBABILITY STATISTICS OF Generate SIGNALS

m	Probability of the generate elements being				P_{err}
	all zero	One 1	Two 1's	Three 1's and more	
2	0.8789	0.1172	0.0039	-	0.00390
3	0.8240	0.1648	0.0110	0.00024	0.01124
4	0.7725	0.2060	0.0206	0.00093	0.02153

being one, two elements being 1 in the same column even decreases. For example, in a column with 4 generate signals, probability of all numbers being 0 is (1 − pr)4, only one element being one is 4pr(1 − pr)3, the probability of two elements being one in the column is 6pr2(1 − pr)2, three ones is 4pr3(1−pr) and probability of all elements being 1 is pr4, where pr is 1/16.

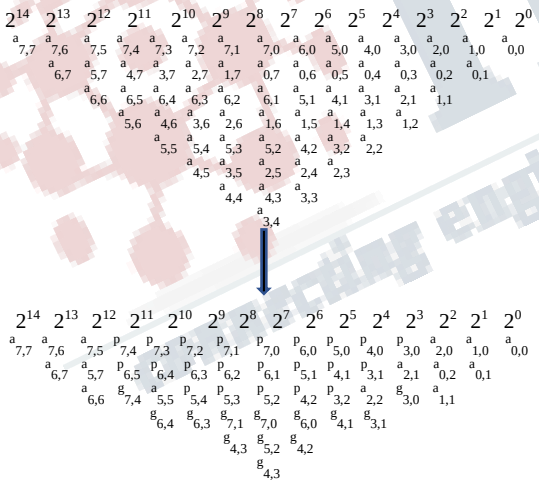

Fig. 1. Transformation of generated partial products into altered partial products.

The probability statistics for a number of generate elements m in each column are given in Table I. Based on Table I, using OR gate in the accumulation of column wise generate elements in the altered partial product matrix provides exact result in most of the cases. The probability of error (P_{err}) while using OR gate for reduction of generate signals in each column is also listed in Table I. As can be seen, the probability of misprediction is very low. As the number of generate signals increases, the error probability

TABLE II
TRUTH TABLE OF APPROXIMATE HALF ADDER

Inputs		Exact outputs		Approximate outputs		Absolute Difference
x1	x2	carry	sum	carry	sum	
0	0	0	0	0✓	0✓	0
0	1	0	1	0✓	1✓	0
1	0	0	1	0✓	1✓	0
1	1	1	0	1✓	1×	1

TABLE III
TRUTH TABLE OF APPROXIMATE FULL ADDER

Inputs			Exact outputs		Approximate outputs		Absolute Difference
x1	x2	x3	carry	sum	carry	sum	
0	0	0	0	0	0✓	0✓	0
0	0	1	0	1	0✓	1✓	0
0	1	0	0	1	0✓	1✓	0
0	1	1	1	0	1✓	0✓	0
1	0	0	0	1	0✓	1✓	0
1	0	1	1	0	1✓	0✓	0
1	1	0	1	0	0×	1×	1
1	1	1	1	1	1✓	0×	1

increases linearly. However, the value of error also rises. To prevent this, the maximum number of generate signals to be grouped by OR gate is kept at 4. For a column having m generate signals, _m/4_ OR gates are used.

B. Approximation of Other Partial Products

The accumulation of other partial products with probability ¼ for $a_{m,n}$ and 7/16 for $p_{m,n}$ uses approximate circuits. Approximate half-adder, full-adder, and 4-2 compressor are proposed for their accumulation. Carry and Sum are two outputs of these approximate circuits. Since Carry has higher weight of binary bit, error in Carry bit will contribute more by producing error difference of two in the output. Approximation is handled in such a way that the absolute difference between actual output and approximate output is always maintained as one. Hence Carry outputs are

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 5, Issue 5, May 2018

approximated only for the cases, where Sum is approximated. In adders and compressors, XOR gates tend to contribute to high area and delay. For approximating half-adder, XOR gate of Sum is replaced with OR gate as given in (2). This results in one error in the Sum computation as seen in the truth table of approximate half-adder in Table II. A tick mark denotes that approximate output matches with correct output and cross mark denotes mismatch

Sum = x1 + x2 Carry = x1 · x2. (2)

In the approximation of full-adder, one of the two XOR gates is replaced with OR gate in Sum calculation. This results in error in last two cases out of eight cases. Carr y is modified as in (3) introducing one error. This provides more simplification, while maintaining the difference between original and approximate value as one. The truth table of approximate full-adder is given in Table III

W = (x1 + x2)

Sum = W ⊕ x3

Carr y = W · x3. (3)

Two approximate 4-2 compressors in [5] produce nonzero output even for the cases where all inputs are zero. This results in high ED and high degree of precision loss especially in cases of zeros in all

bits or in most significant parts of the reduction tree. The proposed 4-2 compressor overcomes this drawback. In 4-2 compressor, three bits are required for the output only when all the four inputs are 1, which happens only once out of 16 cases.

TABLE IV
TRUTH TABLE OF APPROXIMATE 4-2 COMPRESSOR

Inputs				Approximate outputs		Absolute Difference
x1	x2	x3	x4	carry	sum	
0	0	0	0	0✓	0✓	0
0	0	0	1	0✓	1✓	0
0	0	1	0	0✓	1✓	0
0	0	1	1	1✓	0✓	0
0	1	0	0	0✓	1✓	0
0	1	0	1	0×	1×	1
0	1	1	0	0×	1×	1
0	1	1	1	1✓	1✓	0
1	0	0	0	0✓	1✓	0
1	0	0	1	0×	1×	1
1	0	1	0	0×	1×	1
1	0	1	1	1✓	1✓	0
1	1	0	0	1✓	0✓	0
1	1	0	1	1✓	1✓	0
1	1	1	0	1✓	1✓	0
1	1	1	1	1×	1×	1

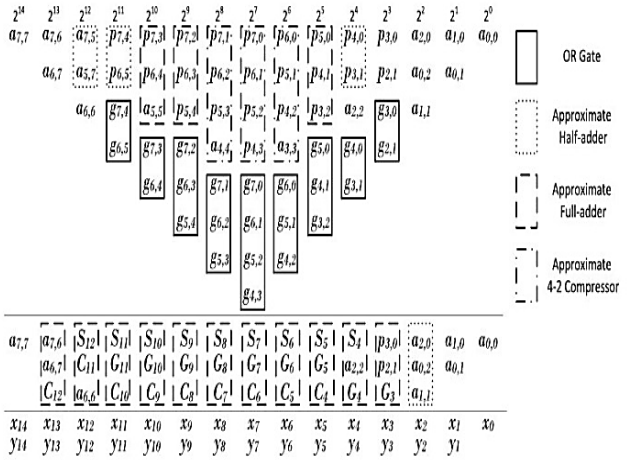

Fig. 2. Reduction of altered partial products.

This property is taken to eliminate one of the three output bits in 4-2 compressor. To maintain minimal error difference as one, the output “100" (the value of 4) for four inputs being one has to be replaced with outputs “11" (the value of 3). For Sum computation, one out of three XOR gates is replaced. With OR gate. Also, to make the Sum corresponding to the case where all inputs are ones as one, an additional circuit x1 · x2 · x3 · x4 is added to the Sum expression. This results in error in five out of 16 cases. Carry is simplified as in (4). The corresponding truth table is given in Table IV

W1 = x1 · x2

W2 = x3 · x4

Sum = (x1 ⊕ x2) + (x3 ⊕ x4) + W1 · W2

Carr y = W1 + W2. (4)

Fig. 2 shows the reduction of altered partial product matrix of 8 ×8 approximate multiplier.

Two stages are required to produce sum and carry outputs of vector merge addition step. In order to reduction of generate signals from columns 3 to 11, four 2-input OR gates, four 3-input OR gates, and one 4-input OR gates are required. Gi corresponding to the column i with 2i are labeled on the resultant signals of OR gates. 3 approximate half-adders, 3 approximate full-adders and 3 approximate compressors are required in the first stage to produce Sum and Carry signals for reducing other partial products. Si and Ci corresponding to column i.e. The elements in the second stage are reduced using 1 approximate half-adder and 11 approximate full-adders producing final two operands xi and yi to be fed to ripple carry adder for the final computation of the result.

C. Two Variants of Multipliers

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 5, Issue 5, May 2018

Two variants of multipliers are proposed. In the first case (Multiplier1), approximation is applied in all columns of partial products of n-bit multiplier, whereas in Multiplier2, approximate circuits are used in n − 1 least significant columns.

III. RESULTS AND DISCUSSION

In this section explained results area, power and delay about designed approximate multiplier using ripple carry adder and carry save adder. And compare with area and delay of both the adder circuit. The final result is Gaussian filter design using approximate multiplier circuit using ripple carry adder.

Ripple carry Adder:

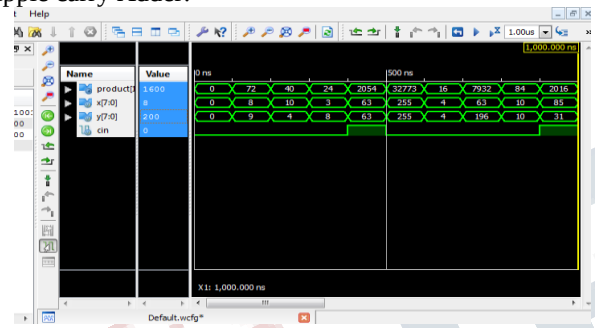

Fig. 3. Simulation result of approximate multiplier using ripple carry adder

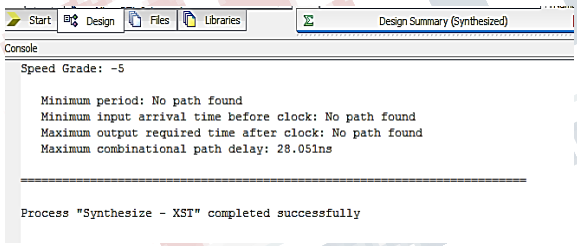

Fig. 4. Delay of approximate multiplier using ripple carry adder

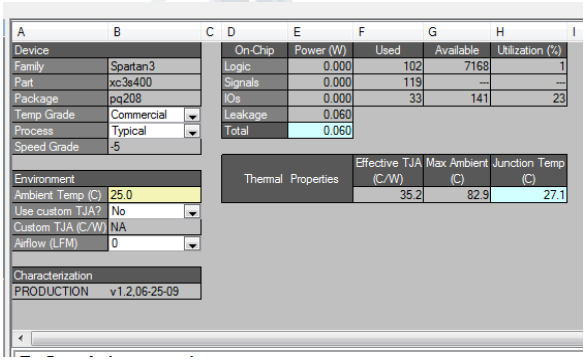

Fig. 5. Power dissipation

Carry save adder:

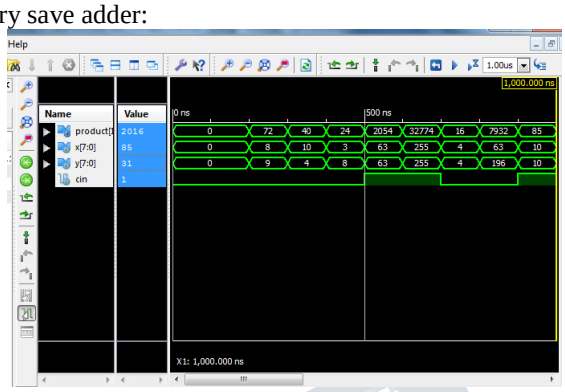

Fig. 6. Simulation result of approximate multiplier using carry save adder

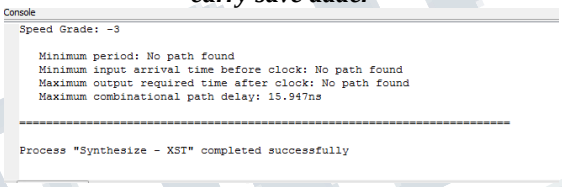

Fig. 7. Delay of approximate multiplier using carry save adder

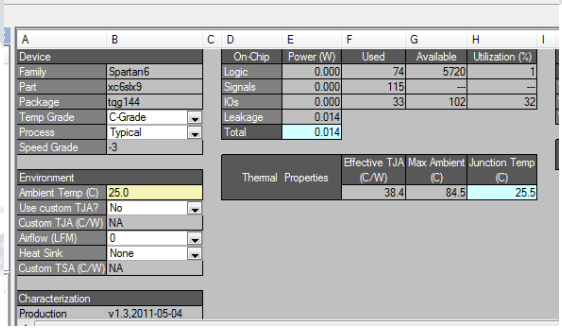

Fig. 8. Power dissipation

IV. APPLICATION—IMAGE PROCESSING

The Gaussian filter is broadly used in image processing application for reducing Gaussian noise [13]. The gaussian filter is better at conserving edge structures than the arithmetic filter. Gray scale images along with Gaussian noise i.e. two 16- bits per pixel are considered. 3 × 3 gaussian filter is used, where each pixel of noise image is replaced with geometric mean of 3 × 3 block of adjacent pixels placed about it. The algorithms are coded and executed using MATLAB. Exact and inexact 16-bit multipliers are used to analyze multiplication between 16-bit pixels.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 5, Issue 5, May 2018

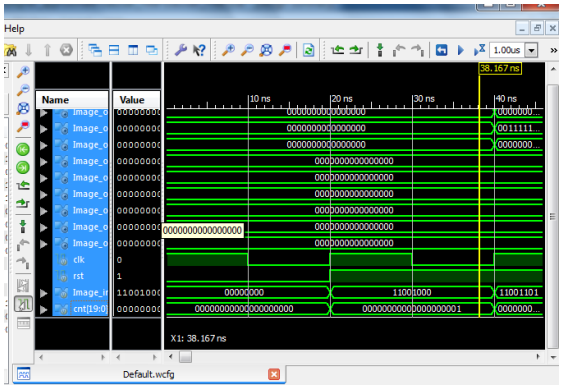

Fig. 9. Simulation result Gaussian filter

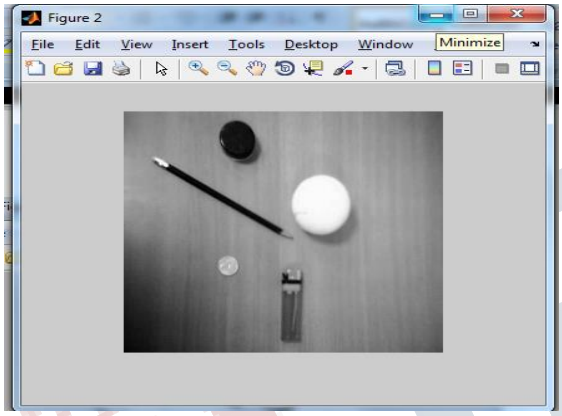

Fig. 9a. Original image (PSNR value is 32.5482).

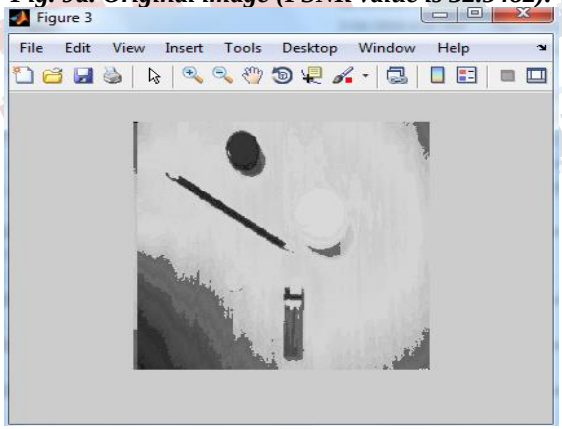

Fig. 9b. Noise reduced image using Gaussian filter (PSNR value is 55.1413).

V. CONCLUSION

In this brief, to propose efficient inexact multipliers, the partial products are produced by using generate and propagate signals. In inexact multiplier, basic OR gate is used for altering generate partial products. Inexact half-adder, full-adder, and 4-2 compressor are used for reducing the remaining partial products. Two variations of inexact multipliers are proposed, where carry save adder achieves substantial reduction in area and power consumption compared with ripple carry adder. The proposed inexact multiplier designs can be utilized in most of the applications with negligible loss in the output quality whereas substantial saving in terms ofpower and area.

REFERENCES

[1] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput.- Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.

[2] E. J. King and E. E. Swartzlander, Jr., "Data-dependent truncation scheme for parallel multipliers," in Proc. 31st Asilomar Conf. Signals,
Circuits Syst., Nov. 1998, pp. 1178–1182.

[3] K.-J. Cho, K.-C. Lee, J.-G. Chung, and K. K. Parhi, "Design of low-error fixed-width modified booth multiplier," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 12, no. 5, pp. 522–531,
May 2004.

[4] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, "Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[5] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and
analysis of approximate compressors for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.

[6] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processing and classification applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

[7] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, "Design-efficient approximate multiplication circuits through partial
product perforation," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 10, pp. 3105–3117, Oct. 2016.

[8] P. Kulkarni, P. Gupta, and M. D. Ercegovac, "Trading accuracy for power in a multiplier architecture," J. Low Power Electron., vol. 7, no. 4, pp. 490–501, 2011.

[9] C.-H. Lin and C. Lin, "High accuracy approximate multiplier with error correction," in Proc. IEEE 31st Int. Conf. Comput. Design, Sep. 2013, pp. 33–38.

[10] C. Liu,J. Han, and F. Lombardi, "A low-power, high-performance Approximate multiplier with configurable partial error recovery," in Proc. Conf. Exhibit. (DATE), 2014, pp. 1–4.

[11] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan, "MACACO: Modeling and analysis of circuits for approximate computing," in Proc.
IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Oct. 2011, pp. 667–673.

[12] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol. 63,
no. 9, pp. 1760–1771, Sep. 2013.

[13] S. Suman et al., "Image enhancement using geometric mean filter and gamma correction for WCE iamges," in Proc. 21st Int. Conf., Neural Inf. Process. (ICONIP), 2014, pp. 276–283.