IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

LOW LATENCY AND LESS COMPLEXITY OF APPROXIMATE MULTIPLIER

BURRA SUDHIR KUMAR[1], Y.REDDY PRASAD REDDY[2]
[1]Associate Professor, Dept of ECE, Princeton College of Engineering and Technology, Ghatkesar, T.S, India.
[2]Associate Professor, Dept of ECE, Princeton College of Engineering and Technology, Ghatkesar, T.S, India.

Abstract- **Approximate multiplier uses simpler structure to generate partial products. In this approximate multiplier can reduce hardware overhead. Approximate computing has received significant attention as a promising strategy to decrease the latency of inherently error-tolerant applications. Hardware approximation mainly targets arithmetic units, e.g. adders and multipliers. This brief deals with a new design approach for approximation of multipliers. The partial products of the multiplier are altered to introduce varying probability terms. Logic complexity of approximation is varied for the accumulation of altered partial products based on their probability. The proposed approximation is utilized 16-bit multiplier.. . In this proposed system we use SQRT CSLA using CBL in order add final step of two rows of partial products. The proposed structure is little bit faster than the previous system. Internal structure of the proposed area-efficient carry select adder is constructed by sharing the common Boolean logic term. They are also found to have better precision and less complexity when compared to existing approximate multiplier designs. The proposed multiplier designs can be used in applications with minimal loss in output quality while saving significant area.**

I. INTRODUCTION

In applications like sight and sound flag preparing and information mining which can endure blunder, correct figuring units are not generally important. They can be supplanted with their rough partners. Research on rough figuring for mistake tolerant applications is on the ascent. Adders and multipliers frame the key segments in these applications. In [1], rough full adders are proposed at transistor level and they are used in advanced flag preparing applications. Their proposed full adders are utilized as a part of amassing of halfway items in multipliers. To diminish gear multifaceted nature of multipliers, truncation is comprehensively used in settled width multiplier designs. By then an enduring or variable change term is added to compensate for the quantization screw up introduced by the truncated part [2], [3]. Figure strategies in multipliers base on accumulating of midway things, which is critical in regards to control usage. Softened show multiplier is executed up [4], where the base basic bits of data sources are truncated, while confining midway things to diminish gear multifaceted nature. The proposed multiplier in [4] saves few snake circuits in partial thing gathering. In [5], two designs of inaccurate 4-2 compressors are presented additionally, used as a piece of midway thing diminish tree of four varieties of 8 × 8 Dadda multiplier. The critical drawback of the proposed compressors in [5] is that they give nonzero yield for zero regarded inputs, which, all things considered, impacts the mean relative slip-up (MRE) as discussed later. The construed design proposed in this brief vanquishes the present drawback.

This prompts better precision. In static part multiplier (SSM) proposed in [6], m-bit areas are gotten from n-bit operands in perspective of driving 1 bit of the operands. By then, m × m increment is performed as opposed to n × n duplication, where m<n. Midway thing gap (PPP) multiplier in [7] neglects k dynamic midway things starting from jth position, where j ∈ [0, n-1] and k ∈ [1, min(n-j, n-1)] of a n-bit multiplier. In [8], 2 × 2 estimated multiplier in perspective of modifying an area in the Karnaugh plot proposed and used as a building square to create 4 × 4 and 8 × 8 multipliers. In [9], mixed up counter layout has been proposed for use in charge successful Wallace tree multiplier. Another evaluated wind is presented in [10] which is utilized for midway thing storing up of the multiplier. For 16-bit assessed multiplier in [10], 26% of diminishment in control is master stood out from revise multiplier. Gauge of 8-bit Wallace tree multiplier on account of voltage over-scaling (VOS) is discussed in [11]. Cutting down supply voltage makes routes fail to meet put off restrictions inciting botch. Past manages method of reasoning unconventionality diminishment revolve around clear utilization of harsh adders and compressors to the midway

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

things. In this short, the midway things are adjusted to give terms different probabilities. Probability estimations of the changed inadequate things are dismembered, which is trailed by think estimation. Streamlined number juggling units (half-snake, full-snake, and 4-2 compressor) are proposed for estimation.

The math units are decreased in versatile quality, and also mind is moreover taken that error regard is cared for low. While foundational estimation helps in achieving better exactness, decreased method of reasoning multifaceted nature of gathered math units eats up less power and zone. The proposed multipliers beats the present multiplier traces the extent that zone, power, and botch, and achieves better zenith banner to upheaval extent (PSNR) values in picture taking care of use.

II. PROPOSED ARCHITECTURE

Implementation of multiplier comprises three steps: generation of partial products, partial products reduction tree, and finally, a vector merge addition to produce final product from the sum and carry rows generated from the reduction tree. Second step consumes more power. In this brief, approximation is applied in reduction tree stage.

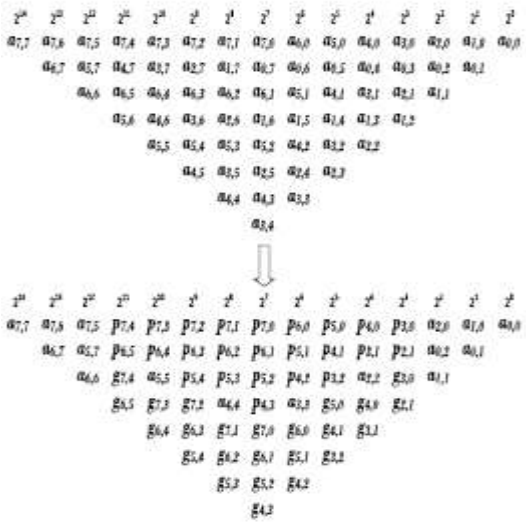


Fig.1. Transformation of generated partial products into altered partial products.

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

Table.1.Probability statistics of generated signals

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.2. Truth table of Approximate Half Adder

A 8-bit unsigned1 multiplier is used for illustration to describe the proposed method in approximation of multipliers. Consider two 8-bitunsigned input operand $\alpha = \sum^{7}_{m=0} \alpha_m 2^m$ and $\beta = \sum^{7}_{n=0} \alpha_n 2^n$. The partial product $a_{m,n} = \alpha_m \cdot \beta_n$ in Fig. 1 is the result of AND operation between the bits of α_m and β_n .From statistical point of view, the partial product $a_{m,n}$ has a probability of 1/4 of being 1. In the columns containing more than three partial products, the partial products $a_{m,n}$ and $a_{n,m}$ are combined to form propogate and generate signals as given in (1). The resulting propogate and generate signals form altered partial products $p_{m,n}$ and $g_{m,n}$. From column 3 with weight 2^3 to column 11 with weight 2^{11}, the partial products $a_{m,n}$ and $a_{n,m}$ are replaced by altered partial products $p_{m,n}$ and $g_{m,n}$. The original and transformed partial Product matrices are shown in Fig. 1.

$$p_{m,n} = a_{m,n} + a_{n,m}$$
$$g_{m,n} = a_{m,n} * a_{n,m} \quad \_(1)$$

The probability of the altered partial product $g_{m,n}$ being one is1/16, which is significantly lower than 1/4 of $a_{m,n}$. The probability of altered partial product $p_{m,n}$ being one is 1/16 + 3/16 + 3/16 =7/16, which is higher than $g_{m,n}$. These factors are considered, while applying approximation to the altered partial product matrix.

A. Approximation of Altered Partial Products gm, n:

The accumulation of generate signals is done column wise. As each element has a

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

probability of 1/16 of being one, two elements being 1in the same column even decreases. For example, in a column with4 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 $6pr^2(1 - pr)^2$, three ones is $4pr^3(1 - pr)$ and probability of all elements being 1 is pr^4, where pr is 1/16.The probability statistics for a number of generate elements. m in each column are given in Table I.

B. Approximation of Other Partial Products

The gathering of other incomplete items with likelihood 1/4 for $a_{m,n}$ and 7/16 for $p_{m,n}$ uses approximate circuits. Approximate half-viper, full-snake, and 4-2 compressor are proposed for their aggregation.

Approximate half-viper, full-snake, and 4-2 compressor are proposed for their aggregation. Carry and Sum are two yields of these approximate circuits. Since Carry has higher weight of parallel piece, blunder in Carry bit will contribute more by delivering mistake contrast of two in the Output. Estimate is taken care of such that the supreme contrast between real yield and rough yield is constantly kept up as one. Henceforth Carry yields are approximated just for the cases, where Sum is approximated. In adders and compressors, XOR entryways have a tendency to add to high region and deferral. For approximating half-viper, XOR door of Sum is replaced with OR entryway as given in (2). This outcomes in a single blunder in the Sum calculation as found in reality table of surmised half-snake in Table 2. A tick check indicates that surmised yield matches with remedy yield and cross stamp signifies bungle.

$$\text{Sum} = x1 + x2$$
$$\text{Carry} = x1 * x2. \quad \_ (2)$$

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. Carry 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 3.

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

Table.3. Truth Table of Approximate Full Adder

$$W = (x1 + x2)$$
$$\text{Sum} = W * x3$$
$$\text{Carry} = W * x3 \quad \_ (3)$$

Two surmised 4-2 compressors in [5] deliver nonzero yield notwithstanding for the situations where all information sources are zero. This outcome in high ED and high level of accuracy misfortune particularly in instances of zeros in all bits or in most critical parts of the lessening tree. The proposed 4-2 compressor conquers this drawback. In 4-2 compressor, three bits are required for the yield just when all the four sources of info are 1, which happens just once out of 16 cases.

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

Table.4. Truth table of Approximate 4-2 comparator

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

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).

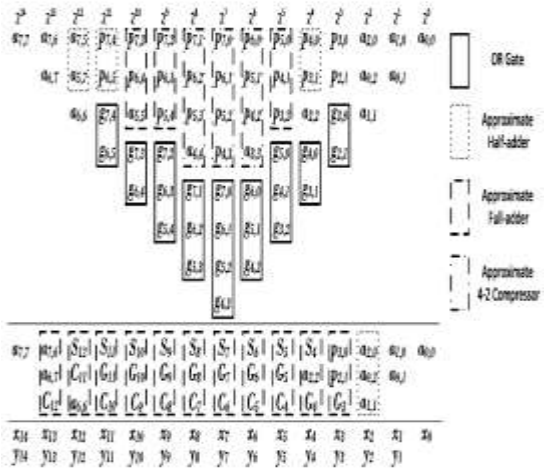


Fig.2. Reduction of altered partial products.

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 $x1 \cdot x2 \cdot x3 \cdot x4$ is added to the Sum expression. This results in error in five out of 16 cases. Carry is simplified assign (4). The corresponding truth table is given in Table 4.

$$W1 = x1 \cdot x2$$
$$W2 = x3 \cdot x4$$
$$Sum = (x1 * x2) + (x3 * x4) + W1 \cdot W2$$
$$Carry = W1 + W2 \quad \_(4)$$

Fig. 2 demonstrates the diminishment of modified fractional item network of 8 ×8 inexact multiplier. It requires two phases to deliver entirety and carry yields for vector consolidate expansion step. Four 2-information OR entryways, four 3-information OR doors, and one 4-info OR entryways are required for the reduction of create signals from segments 3 to 11. The resultant signals of OR doors are named as G_i comparing to the section I with weight 2_i. For decreasing other fractional items, 3 estimated half-adders, 3 surmised full-adders, and 3 rough compressors are required in the principal stage to deliver Sum and Carry signs, S_i and C_i comparing to segment I. The components in the second stage are decreased utilizing 1 rough half-snake and 11 inexact full-adders delivering last two operands xi and yi to be nourished to swell Carry viper for the last calculation of the outcome.

PROPOSED SQRT CSLA USING COMMON BOOLEANLOGIC

To remove the duplicate adder cells in the conventional CSLA, an area efficient SQRT CSLA is proposed by sharing Common Boolean Logic (CBL) term. While analysing the truth table of single bit full adder, results show that the output of summation signal as carry-in signal is logic “0” is inverse signal of itself as carry-in signal is logic “1”. It is illustrated by circles in Table 5.

Table- 5
Truth Table Of Single Bit Full Adder, Where The Upper Half Part Is The Case Of Cin=0 And The Lower Half Part Is The Case Of CIN=1

C_{in}	A	B	S_0	C_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table.5. Single Bit Full Adder

To share the Common Boolean Logic term, we only need to implement a XOR gate and one INV gate to generate the summation pair. And to generate the carry pair, we need to implement one OR gate and one AND gate. In this way, the summation and carry circuits can be kept parallel.

This method replaces the Binary to Excess-1 converter add one circuit by common Boolean logic. As compared with modified SQRT CSLA, the proposed structure is little bit faster. Internal structure of proposed CSLA is shown in Fig. 3.

IJMTARC – VOLUME – VI – ISSUE – 24, OCT-DEC, 2018 ISSN: 2320-1363

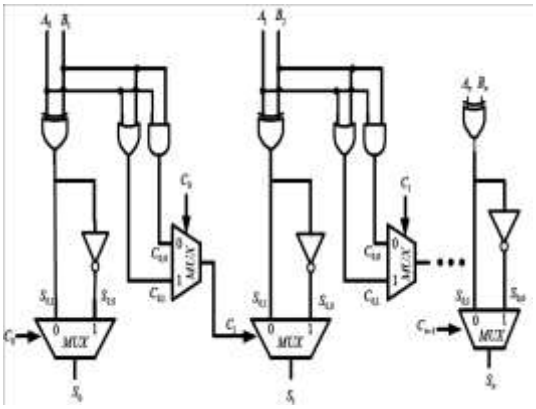


Fig.3 Internal structure of the proposed area-efficient carry select adder is constructed by sharing the common Boolean logic term.

In the proposed SQRT CSLA, the transistor count is trade-off with the speed in order to achieve lower power delay product. Thus the proposed SQRT CSLA using CBL is better than all the other designed adders. Fig. 4 shows the block diagram of Proposed SQRT CSLA.

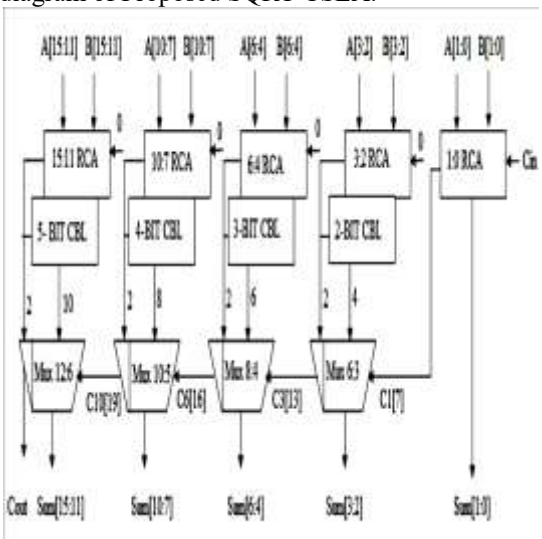
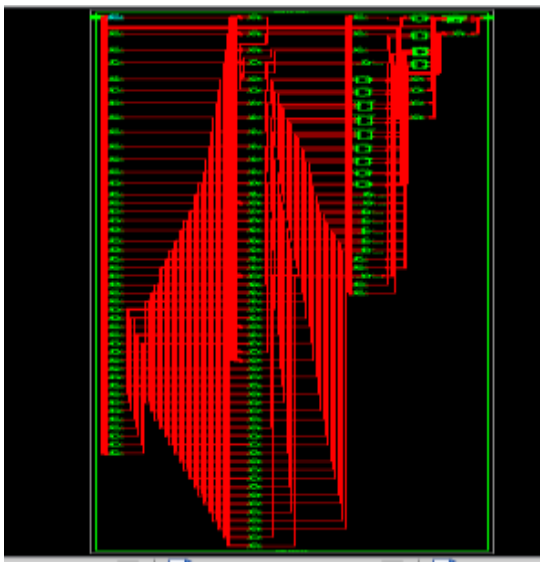


Fig.4. 16-Bit Proposed SQRT CSLA using Common Boolean Logic.

III. RESULTS



RTL Schematic

Technological schematic

Simulation output

IV. CONCLUSION

In this brief, to propose efficient approximate multipliers, partial products of the

multiplier are modified using generate and propagate signals. The generate and propogate signals are generated based on some equations which we are discussed in this paper. Approximation is applied using simple OR gate for altered generate partial products. Approximate half-adder, full-adder, and 4-2 compressor are proposed to reduce remaining partial products. After applying the approximation process we get two rows of partial product at this we use SQRT CSLA using CBL in order add this two rows of partial products. The proposed structure is little bit faster than the previous system. Internal structure of the proposed area-efficient carry select adder is constructed by sharing the common Boolean logic term. They are also found to have better precision when compared to existing approximate multiplier designs. The proposed multiplier designs can be used in applications with minimal loss in output quality while saving significant power 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 errorcorrection," 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-performanceapproximate multiplier with configurable partial error recovery," in *Proc.Conf. Exhibit. (DATE)*, 2014, pp. 1–4.