Indo-Iranian Journal of Scientific Research (IIJSR)
Volume 2, Issue 1, Pages 01-08, January-March 2018

Design of Approximate Dadda Multiplier for Low Power and Efficient Area

Dr.V.M.Senthil Kumar[1] and Ms.G.Anish Girija[2]

[1]*Associate Professor, Department of Electronics and Communication Engineering, Vivekanandha College of Engineering for women, Tamilnadu, India.*
[2]*PG Scholar, Department of Electronics and Communication Engineering, Vivekanandha College of Engineering for women, Tamilnadu, India.*

Article Received: 27 November 2017 *Article Accepted: 30 January 2018* *Article Published: 04 February 2018*

ABSTRACT

The design complexity with an increase in performance and power efficiency for error resilient applications can decrease approximate computing. This approach for an approximation of multipliers deals with recent design. To begin of changing probability terms in the design the partial products of the approximate multiplier are also changed. Based on their probability logic complexity the approximation is varied for the accumulation of altered partial products. The proposed multiplier is designed in two variants of 16-bit multipliers. Synthesis results shows that the two designed multipliers produces power savings of about 72% and 38% when it is compared to an existing multiplier. When compared to existing approximate multiplier they have better precision and the Mean relative error figures are as low as 7.6% and 0.02% for the proposed approximate multipliers which are better than the earlier works. Performance of the proposed multipliers is evaluated in an image processing application, thus one of the planned models achieves the maximum peak signal to noise ratio.

Keywords: Design complexity and Logic complexity.

1. INTRODUCTION

In applications such as multimedia signal processing and data mining which are capable to tolerate error, exact computing units are not necessary. They can be a substitute for their approximate counterparts. Research on approximate computing for error-tolerant applications is significant increases. Adders and multipliers form the key components in these applications. In approximate full adders are designed at transistor level and they make use of in digital signal processing applications. Their planned full adders are used in the accumulation of partial products in multipliers.

To reduce hardware complexity of multipliers, truncation is broadly employed in fixed-width multiplier designs. Then a constant or variable correction term is included to balances quantization error that is introduced by the truncated part. The Approximation method in multipliers deals with the accumulation of partial products, in terms of power consumption. Broken array multiplier is implemented where the least significant bits of inputs are truncated while forming partial products to reduce hardware difficulty. The proposed multiplier design save few adder circuits in partial product accumulation.

In, two designs of approximate 4-2 compressors they use a partial product reduction tree of four variants of 8 × 8 Dadda multiplier. The major drawback of the proposed compressor, when it gives non-zero output for zero-valued inputs, which mainly have an effect on the mean relative error (MRE). The approximate design proposed to the point overcomes the existing trouble. This leads to better precision. In static segment multiplier (SSM) proposed design includes *m*-bit segments that are obtained from *n*-bit operands based on leading 1 bit of the operands. Then, *m*×*m* multiplication is performed as an alternative of *n*×*n* multiplication, whereas the term *m is lesser than n*. In 2 × 2 approximate multipliers based on modifying an entry in the Karnaugh map is proposed and used as a building block to make 4 × 4 and 8 × 8 multipliers. The inaccurate counter design has been proposed for the use of power efficient Wallace tree multiplier.

A new approximate adder is presented in which is utilized for partial product accumulation of the multiplier. For 16-bit approximate multiplier, 26% of fall in power is consummate as compared to the exact multiplier. The Logic complexity deals with a straight-forward application of approximate adders and compressors to the partial products. In this brief, the partial products are transformed to initiate terms with different probabilities. Probability statistics of the altered partial products are analyzed, which is followed by systematic approximation. Simplified arithmetic units such as half-adder, full-adder, and 4-2 compressor are planned for approximation. The arithmetic units are not only used to reduce the complexity, but care is also taken to maintain the error value low. While systemic approximation is useful in achieving better accuracy, the reduced logic complexity of approximate arithmetic units consumes low power and efficient area. The proposed multipliers perform the existing multiplier designs in terms of area, power, and error, and achieves highest peak signal to noise ratio (PSNR) values in image processing application.

Error distance (ED) can be defined as the arithmetic distance between a correct output and approximate output for a given input. In approximate adders are evaluated and normalized ED (NED) is proposed as equivalent to invariant metric independent of the size of the approximate circuit. Also, traditional error analysis, MRE is found for existing and proposed multiplier design. The detailed outcome for the analysis of error metrics in proposed and existing approximate multipliers are explained in the further sections. The proposed multipliers are utilized in image processing application.

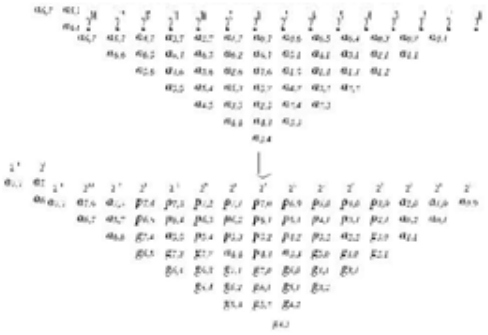

Fig. 1. Transformation of generated partial products into altered Partial Products.

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

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

2. PROPOSED SYSTEM

Implementation of multiplier includes three steps: generation of partial products, partial products reduction tree and at last, a vector merge addition is used to turn out final product from the sum and carry rows generated from the reduction tree. From a statistical point of view, the partial product obtain a probability of 1/4 of being 1. In the columns containing additional three partial products, the partial products $a_{m,n}$ and $a_{n,m}$ are joint to form propogate and generate signals. The resultant 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 partial products $p_{m,n}$ and $g_{m,n}$. The original and transformed partial product matrices are shown in Fig. 1.

$$P_{m,n}, = a_{m,n} + a_{n,m}$$
$$G_{m,n} = a_{m,n} \cdot a_{n,m} \qquad (1)$$

The probability of the transformed partial product $g_{m,n}$ is given as 1/16, which is considerably lower than 1/4 of $a_{m,n}$. The probability of transformed partial product $p_{m,n}$ is given as 1/16 + 3/16 + 3/16 = 7/16, which is greater than $g_{m,n}$. These factors are considered while applying an approximation to the transformed partial product matrix.

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

The accumulation of *generate* signals is completed through column-wise. As each element has a probability of 1/16 of being one, two elements being 1 in the similar column yet decreases. By using Table I, with OR gate in the accumulation of column-wise *generate* elements in the transformed partial product matrix provides exact result in most of the cases. The probability of fault while by means of OR gate for a decrease of *generate* signals in every column is also listed in Table I. The probability of misprediction is extremely low. As the amount of *generate* signals increases, the error probability increases linearly. However, the value of error also increases. To avoid this, the maximum number of *generate* signals to be grouped by OR gate is kept at 4.

B. Approximation of Other Partial Products

The accumulation of previous partial products with probability 1/4 for $a_{m,n}$ and 7/16 for $p_{m,n}$ mostly uses approximate circuits. Approximate half-adder, full-adder, and 4-2 compressor are designed for their accumulation. *Carry* and *Sum* is two outputs of these approximate adder circuit. Since *Carry* has a greater weight of binary bit, error in *Carry* bit will contribute further by producing error dissimilarity of two in the output. The Approximation is handled in such a way that the complete difference between actual output and approximate output is always maintained to be 1. Hence *Carry* outputs are approximated only for the certain cases, where *Sum* is approximated. In adders and compressors, XOR gates are likely to contribute to a high area and high delay. For approximating half-adder, XOR gate of *Sum* is altered with OR gate as given in (2). This gives the result of 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 exact output and cross mark denote mismatch.

$$\text{Sum} = x1 + x2$$
$$\text{Carry} = x1 \cdot x2. \qquad (2)$$

In the approximation of full-adder, one of the two XOR gates is altered with OR gate in *Sum* computation. This results in the error in last two cases out of eight cases. *Carry* is customized in introducing an error. This provides high simplification while maintaining the difference between new and approximate value as one. The truth table of approximate full-adder is given in Table III.

$$W = (x1 + x2)$$
$$\text{Sum} = W \oplus x3$$
$$\text{Carry} = W \cdot x3. \qquad (3)$$

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

Table V

Synthesis Results of Exact, Existing, and Proposed Approximate Multipliers

Multiplier Type	Area (μm^2)	Delay (ns)	Power (μW)	PDP (fJ)	APP ($\mu m^2 \cdot \mu W)(10^5$)
Exact	4859.28	0.68	1776.49	1208.01	86.32
Multiplier1	2158.56	0.47	503.15	236.48	10.86
Multiplier2	3319.20	0.66	1102.03	727.34	36.57
ACM1 [5]	2871.72	0.4	435.31	174.12	12.50
ACM2 [5]	3782.16	0.63	1250.70	787.94	47.30
SSM [6]	3953.88	0.69	1225.29	845.45	48.44
PPP [7]	4547.52	0.64	1570.79	1005.31	71.43
UDM [8]	3938.00	0.67	1318.51	883.40	51.92

To sustain minimal error variation as one, the output "100" for four inputs being one has to be substitute with outputs "11". For *Sum* computation, one out of three XOR gates is altered with OR gate. And also to make the *Sum* equivalent to the case where all inputs are ones as one, an additional circuit $x\,1 \cdot x\,2 \cdot x\,3 \cdot x\,4$ is added to the *Sum* expression. This results in the error in five out of 16 cases. *Carry* is simplified as in (4). The corresponding truth table is given in Table IV. Fig. 2 shows the reduction of altered partial product matrix of 8 × 8 approximate multipliers. It requires two stages to make sum and carry outputs for vector merge addition process. Four 2-input OR gates, four 3-input OR gates, and one 4-input OR gates are necessary for the decrease of *generate* signals from columns 3 to 11. The resulting signals of OR gates are categorized as G_i corresponding whose column i with weight 2^i.

$$W1=x1.x2$$

$$W2=x3.x4$$

In addition to reducing other partial products, 3 approximate half-adders, 3 reasonably accurate full-adders, and 3 approximate compressors are essential in the first stage to create *Sum* and *Carry* signals, S_i and C_i equivalent to column *i*.

3. RESULTS AND DISCUSSION

All approximate multipliers are considered for $n = 16$. The multipliers are simulated in Verilog and the design synthesized with Synopsys Design Compiler and a TSMC 65 nm standard cell library with a temperature range of 25 °C and supply voltage 1 V. We obtain area, delay, dynamic power and leakage power from the Synopsys dc reports. Multiplier1 applies approximation in all columns, whereas in Multiplier2, an approximation is useful in 15 least significant columns during partial product drop. The altered partial products are generated and compressed using half-adder, full-adder, and 4-2 compressor structures to form final partial products.

The overall efficiency of the proposed multipliers is compared with existing approximate multipliers. Inexact compressor design 2 is used to design compressor based on multipliers where all columns are approximated in ACM1 and ACM2, where only 15 least significant columns are approximated. SSM for $m = 12$ and $n = 16$ is designed for implementation. PPP design for $j = 2$, $k = 2$ is designed and implemented under Dadda tree structure. In the partial product matrix of 16-bit Under Designed Multiplier (UDM) includes approximate 2 × 2 partial products accumulated collectively with exact carry save adders. Exhaustive error analysis of the approximate multipliers is completed using MATLAB. The Exact 16-bit multiplier is designed using Dadda tree structure.

The design in terms of area, delay, power, Power Delay Product (PDP), and Area Power Product (APP) are compared and shown in Table V. If high approximation can be tolerated for saving more power, Multiplier1 and ACM1 are the candidates to be considered. It can be seen that Multiplier1 has a better APP, whereas ACM1 has a better PDP. Multiplier 2 offers 32% area savings and 38% power savings, over the exact multiplier. ACM2 provides 22% and 30% area and power savings, respectively. SSM has 19% area and 31% power savings over

accurate multiplier. The Perforated multiplier has 6% and 12% area and power savings, respectively. UDM provides 19% and 26% area and power savings. Multiplier2 has one order of lower MRE than ACM2, two orders of lower MRE than UDM, 73% lower MRE than PPP, and 62% lower MRE than SSM. NED of Multiplier2 outperforms all approximate multipliers apart from ACM2. ACM2 exhibits 10% lower NED than Multiplier2. Multiplier2 produces large ED relative to ACM2. However, lower MRE indicates that Multiplier2 has lesser relative error values. MRE distribution of 16-bit versions of Multiplier1 and Multiplier2 is shown in Fig. 3. All possible outputs varying from 0 to 65535^2 are categorized into 255 intervals. MRE of Multiplier2 is considerably low at higher product values, as exact units are used in most of the multiplier.

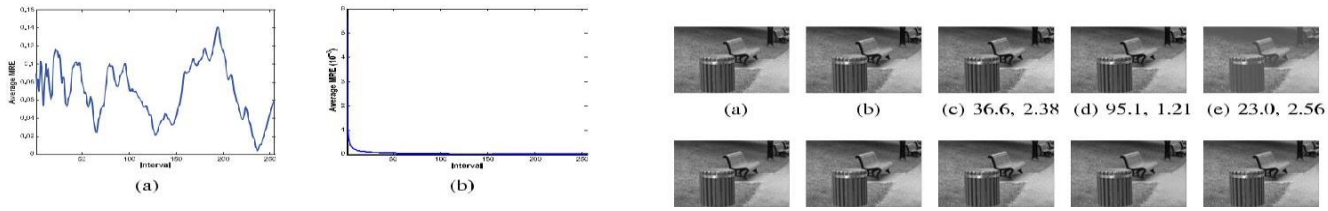

Fig.3. MRE distribution of (a) Multiplier1 and (b) Multiplier2.

4. APPLICATION—IMAGE PROCESSING

To reduce Gaussian noise geometric mean filter is widely used in image processing applications. The geometric mean filter is better at preserving edge features than the arithmetic mean filter. Two 16-bits per pixel gray scale images with Gaussian noise are considered. 3 × 3 mean filter is used, where each pixel of a noisy image is replaced with the geometric mean of 3 × 3 blocks of adjacent pixels centered about it.

PSNR is based on mean-square error found between resulting image of the exact multiplier and the images generated from approximate multipliers. The Energy requires for an exact and approximate multiplication process while performing geometric mean filtering of the images is established by using Synopsys Primetime. In addition to the exact multiplier is voltage scaled from 1 to 0.85 V (VOS), and its impact on energy consumption and image quality is computed.

The noisy input image and resulting image by using exact and approximate multipliers, with their respective PSNRs and energy savings are evaluated. The Energy required for an exact multiplication process for image-1 and image-2 is 3.24 and 2.62μJ correspondingly. Even though ACM1 has better energy savings compared to Multiplier1, Multiplier1 has significantly higher PSNR compared to ACM1. Multiplier2 shows the finest PSNR between all the approximate designs. Multiplier2 has better energy savings, compared to ACM2, PPP, SSM, UDM, and VOS. The intensity of image-1 being mostly on the lower end of the histogram causes reduced performance of ACM multipliers. As the switching activity impacts most important part of the design in VOS, PSNR values are affected.

5. CONCLUSION

In this brief, to propose efficient approximate multipliers, partial products of the multiplier are customized using *generate* and *propagate* signals. The Approximation is applied using simple OR gate for transformed *generate* partial products. Approximate half-adder, full-adder, and 4-2 compressor are planned to decrease remaining partial products. Two variants of approximate multipliers are proposed, where the approximation is useful in all *n* bits in Multiplier1 and only in n-1 least significant part in Multiplier2. Multiplier1 and Multiplier2 achieve a significant reduction in area and power consumption compared with exact designs.

The APP savings being 87% and 58% for design 1 and design 2 with respect to exact multipliers, they also perform in APP in comparison with present approximate designs. They are also establishing to have better precision when compared to existing approximate multiplier. The proposed multiplier designs can be used in applications with minimal loss in output quality while saving significant power and area.

REFERENCES

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

[3]. Mahdiani. H. R, 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.

[4] Momeni. A., 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.

[5] Narayanamoorthy.M, Moghaddam. H. A, Liu. Z, Park. T, and Kim. N. S, "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.

[6] Senthilkumar.V.M, Sowmiya.S, "Design A FinFET Based 4-2 Compressor for Arithmetic Operation," Advances in Natural and Applied Sciences., pp. 605-610.

[7] Senthilkumar.V.M, Stella.K, "Design A Full Adder *Based* GDI Logic Using *FinFET* Technology", Advances in Natural and Applied Sciences. 11(6); Pages: 599-604.

[8] Zervakis.G, Tsoumanis.K, 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.