ISSN: 2395-1680 (ONLINE)
DOI: 10.21917/ijme.2021.0184

DESIGN OF ENERGY EFFICIENT APPROXIMATE MULTIPLIERS FOR IMAGE PROCESSING APPLICATIONS

Chinna V. Gowdar[1] and M.C. Parameshwara[2]

[1]*Department of Electronics and Communication Engineering, Rao Bahadur Y Mahabaleswarappa Engineering College, India*
[2]*Department of Electronics and Communication Engineering, Vemana Institute of Technology, India*

Abstract

This research paper presents the design of two 8×8 approximate multipliers based on novel approximate 3:2 and 2:2 compressors. The proposed multipliers are derived based on Wallace multiplier architecture and herein referred to as the proposed 'approximate Wallace multiplier' (AWM). The performance of these proposed AWMs has been assessed and analyzed in terms of 'Design Metrics' (DMs) such as power, delay, 'power-delay-product' (PDP), and area. Further, a performance comparison of AWMs has been carried out against 6 other multipliers designed based on reported approximate 3:2 compressors. To extract these DMs, all the multipliers under consideration have been described using Verilog code and synthesized using Cadence's 'RTL Compiler' (RC) tool using a 180 nm standard cell library. The synthesis results show that the proposed AWMs accomplish an excellent performance in terms of DMs. Further, the AWMs along with other designed Wallace multipliers, based on reported approximate compressors have been compared, under image processing application in terms of 'peak signal-to-noise ratio' (PSNR). The comparison results show that the proposed multipliers have a better PSNR (more than 50 dB).

Keywords:

Approximate Computation, Wallace Multiplier, 3:2 Compressor, Low Power, PDP

1. INTRODUCTION

The next-generation embedded systems need to process a huge amount of computationally intensive data related to multimedia applications such as audio, speech, video, and image processing [4]. The unique feature of these applications is that they are highly resilient to errors and do not need perfect accuracy in computation. This relaxation on accuracy or correctness provides a degree of freedom to use approximate or imprecise circuits in the aforementioned applications [1] [2] [4] [5] [7]-[10]. In such applications, to tradeoff the performance and accuracy, the approximate computation is preferred as a choice over exact computation. The arithmetic systems that handle these multimedia applications generally use, the 'Digital-Signal-Processing' (DSP) blocks as a core to process various arithmetic operations. The most basic operations in such DSP blocks are addition and multiplication. The adder circuit is a critical element, plays an important role to determine the overall accuracy and performance of a system under consideration. Thus, in the context of DSP, the design and optimization of adder and multiplier circuits in terms of energy efficiency always finds an interesting area of research [11].

In the state-of-the-art literature, several approximate multipliers have been reported and analyzed for multimedia applications. In [7], an approximate Baugh-Wooley multiplier is presented and analyzed its performance using a 64-tap low-pass 'finite impulse response' (FIR) filter. A rounding-based approximate multiplier for both signed and unsigned multiplication is presented in [13]. Comparison of Wallace tree and Dadda multiplier architectures in terms of power, delay, PDP, and area have been discussed in [14]. Four unsigned approximate multipliers have been proposed [16] based on novel 4:2 compressors and analyzed their performance in terms of power, delay, 'transistor count' (TC), and PSNR. Several 4:2 approximate compressors have been proposed in [16]. An improved approximate multiplier based on modified 4:2 compressors [16, 17] is presented in [6]. Based on approximate arithmetic circuits, several multipliers have been proposed and analyzed in [1] [2] [4] [8] [22]. Further many approximate 1-bit full adder (FA) cells were discussed and analyzed at different levels of design abstraction [15] [18]–[21] [23].

The rest of this paper is structured as follows. Section 2 details the design of the proposed AWM using novel approximate compressors. Section 3 presents the synthesis environment, simulation results and discussion and an image processing application of proposed multipliers. Section 4 concludes this paper.

2. DESIGN OF PROPOSED WALLACE MULTIPLIER USING NOVEL APPROXIMATE COMPRESSORS

This section discusses the proposed 3:2 and 2:2 compressors and their utilization in the design of an 8×8 AWM.

2.1. PROPOSED COMPRESSORS

This work presents three new 'approximate compressors' (ACs) namely AC_1, AC_2, and AC_3. The AC_1 and AC_2 are 3:2 compressors and AC_3 is a 2:2 compressor. These compressors are derived by modifying the 'truth table' (TT) of 'exact compressors' (ECs) namely EC_1, EC_2, where the EC_1 and EC_2 are exact 3:2 and 2:2 compressors.

Table.1. Truth Table of Exact and proposed approximate 3:2 compressors

Inputs			EC_1		AC_1		AC_2	
A	***B***	C_{in}	***Sum***	C_{out}	Sum_1	C_{out1}	Sum_2	C_{out2}
0	0	0	0	0	1×	0✓	1×	0✓
0	0	1	1	0	1✓	0✓	1✓	0✓
0	1	0	1	0	0×	0✓	0×	0✓
0	1	1	0	1	0✓	0×	0✓	0×
1	0	0	1	0	0×	1×	0×	0✓
1	0	1	0	1	0✓	1✓	0✓	0×

1	1	0	0	1	0✓	1✓	0✓	1✓
1	1	1	1	1	0×	1✓	0×	1✓
ED			-		4	2	4	2

Table.2. Truth Table of Exact and proposed approximate 2:2 compressor

Inputs		***EC*$_2$**		***AC*$_3$**	
X	***Y***	***Sum***	***C*$_{out}$**	***Sum*$_3$**	***C*$_{out3}$**
0	0	0	0	1×	0✓
0	1	1	0	1✓	0✓
1	0	1	0	1✓	0✓
1	1	0	1	0✓	1✓
ED		-		1	0

Table.3. Output Logic Expressions of Proposed ACs

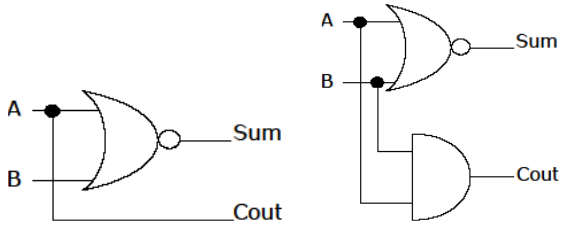
Proposed Compressor	**Output Equations**	
	Sum	***C*$_{out}$**
AC$_1$	$\bar{A}+\bar{B}$	*A*
AC$_2$	$\bar{A}+\bar{B}$	*A*·*B*
AC$_3$	$\bar{A}+\bar{B}$	*A*·*B*

The TT of respective ACs and ECs are shown in Table.1 and Table.2 respectively. The Table.1 consists of 3 inputs: *A*, *B*, and C_{in} and 6 outputs: *Sum*, C_{out}, Sum_1, C_{out1}, Sum_2 and $Cout_2$. The Sum, Sum_1, and Sum_2 are '*Sum*' outputs of EC_1, AC_1, and AC_2 respectively. The C_{out}, C_{out1}, and C_{out2} are 'Carry' outputs of EC_1, AC_1, and AC_2 respectively. The Table.2 consists of 2 inputs *X* and *Y* and 4-outputs: *Sum*, C_{out}, Sum_3 and C_{out3}. The *Sum* and Sum_3 are '*Sum*' outputs of EC_2 and AC_3 respectively. The C_{out} and $Cout_3$ are the '*Carry*' outputs of EC_2 and AC_3 respectively.

The TTs of ACs are derived by using an appropriate number of min-terms in each approximate output. The total number of min-terms and their respective positions are chosen such that: the derived logic equations for a given approximate output should meet the following constraints:

- The 'error distance' (ED) should be moderate.
- The simplified output equation should contain a few 'sum-of-product (SOP)' terms.
- Each SOP term should have only few literals.

The term ED represents the arithmetic distance between the approximate and exact outputs. The simplified output equations for the exact and proposed compressors are listed in Table.3 and their corresponding logic diagrams are shown in Fig.1.



(a) AC_1 (b) AC_2 or AC_3

Fig.1. Logic diagram of proposed approximate compressors

2.2. DESIGN OF APPROXIMATE WALLACE MULTIPLIER

This section presents the design of an *N*×*M* 'Approximate Wallace multiplier' (AWM) using the proposed 'approximate compressors' (ACs). The *N* and *M* represent the size of multiplicand and multiplier respectively. In this research work, the values of *N* and *M* are chosen such that *N*=*M*=8-bits. The Wallace multiplier based on exact compressors (ECs) consists of three blocks [14]:

- Partial product generation (PPG)
- Partial product compression (PPC)
- Carry propagate addition (CPA)

The PPG block generates 64 (=8×8) 'partial products' (PPs) and these PPs are arranged in the form of an array having 8 rows and 15 columns. The PPC block is used to reduce the 8 rows of PPs into 2 rows for final binary addition. The PP reduction is carried out through different stages using compressors.

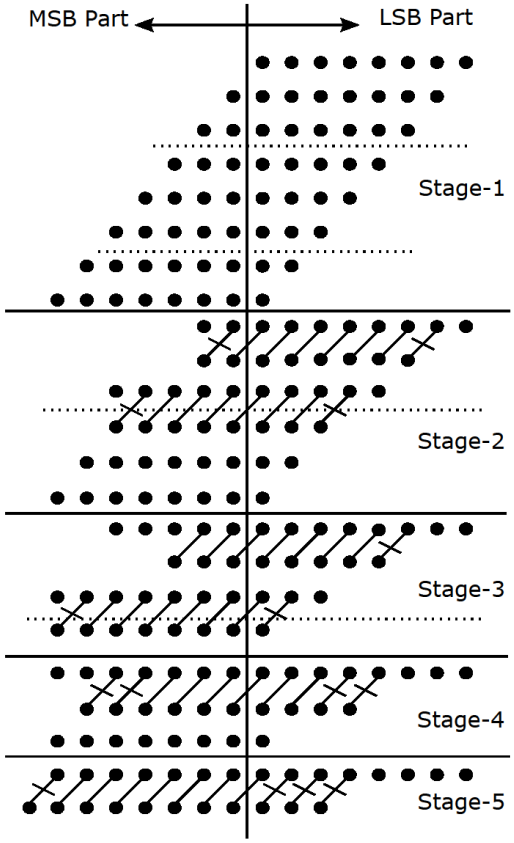


Fig.2. Reduction circuitry of 8×8 Wallace multiplier

The 2 rows of PPs are then added to produce a final product. The process of PP reduction using 3:2 and 2:2 compressors is illustrated in Fig.2. To investigate the impact of proposed *AC*s, two *AWM*s have been designed and analyzed. The *AWM* designed using AC_1 and AC_3 herein referred to as 'Proposed AWM_1' ($PAWM_1$) and using AC_2 and AC_3 is referred to as 'Proposed AWM_2' ($PAWM_2$). The $PAWM_1$ and $PAWM_2$ have been derived based on the reduction circuitry shown in Fig.2. This figure is composed of two parts: 'Most Significant Bits' (MSB) and 'Least Significant Bits' (LSB). On the LSB part, the reduction of PPs (from Stage 1 to Stage 4) and carry propagate addition of final 2

rows (Stage 5) is carried out using approximate 3:2 and 2:2 compressors. The same on the MSB part is implemented using exact compressors. On the LSB part: the AC_1 and AC_3 are used in $PAWM_1$ and the AC_2 and AC_3 are used in $PAWM_2$, respectively.

3. SIMULATION RESULTS AND DISCUSSION

This section discusses the simulation results and performance comparison of proposed multipliers. For performance comparison, a set of AWMs has been designed and utilized. This set of designs herein referred to as 'Designed AWMs' (DAWMs). All DAWMs have been implemented as per the reduction circuitry shown in Fig.2. The approximate compressors required to design DAWMs are herein referred to as 'reported compressors' (RCs). All the RCs have been designed based on the output logic equations derived from the truth tables [18] [15]. The set of output logic equations used to model the RCs are listed in Table.4.

Table.4. Output Logic Expressions of Reported ACs (RCs)

RC	Output Equations		Ref.
	Sum	***C_{out}***	
RC_1	$A \oplus B \oplus C_{in}$	C_{in}	[12]
RC_2	$A \oplus B$	$A \cdot B$	[12]
RC_3	$\bar{A} \cdot \bar{B} + \bar{A} \cdot \bar{C}_{in} + \bar{B}\bar{C}_{in}$	$A \cdot B + A \cdot C_{in} + B \cdot C_{in}$	[12]
RC_4	$\bar{C}_{in} \cdot \bar{A} \oplus \bar{B}$	$B + A \cdot C_{in}$	[15]
RC_5	$\bar{A} \cdot \bar{B} + \bar{A} \cdot \bar{C}_{in}$	$B + A \cdot C_{in}$	[15]
RC_6	$\bar{A} \cdot C_{in} + B \cdot C_{in}$	A	[15]

Table.5. Output Logic Expressions of Exact Compressors (EC)

EC	Output Equations	
	Sum	***C_{out}***
EC_1	$A \oplus B \oplus C_{in}$	$A \cdot B + A \cdot C_{in} + B \cdot C_{in}$
EC_2	$A \oplus B$	$A \cdot B$

Further, the exact multiplier herein referred to as 'exact Wallace Multiplier' (EWM) is also designed and used for performance comparison. The LSB and MSB parts of EWM have been designed using exact compressors EC_1 and EC_2. The set of logic equations used to design these exact compressors are listed in Table.5. To assess the performance of proposed multipliers ($PAWM_1$ and $PAWM_2$), the DMs such as power, delay, PDP, and area are extracted and compared against the EWM and DAWMs.

3.1. SYNTHESIS ENVIRONMENT

To extract these DMs, all the multipliers under consideration have been designed using Verilog RTL codes. To verify the functionality of the multipliers under consideration, a Verilog test bench code is written and simulated using Cadence's 'NCSim' tool. Further, the functionality of all the multipliers under consideration have been verified through a common Verilog test bench code. The RTL codes are then synthesized using Cadence's 'RTL Compiler' (RC) tool using common 'Process-Voltage-Temperature' (PVT) conditions.

The synthesis environment used to extract the DMs is shown in Fig.3. The Verilog RTL code and TSMC 180 nm standard cell library are fed as an input to the synthesis tool. With these inputs, the Cadence's RTL compiler (RC) generates gate level net-list to extract the required DMs. For a fair comparison, all the multipliers under consideration have been described using Verilog RTL codes and synthesized using supply voltage of V_{dd}=1.8V and temperature 27ºC.

Fig.3. Synthesis environment used to extract the DMs

To extract post synthesis DMs, the default input-drive strength and output load have been used. The extracted results from the synthesis are tabulated in Table.6.

Table.6. Performance comparison of designed and proposed approximate Wallace multipliers in terms of DMs

Approximate Multiplier	Power (nW)			Delay (ps)	PDP (fJ)	Area (μm^2)
	Leakage	Dynamic	Total			
EWM	31.978	505595.093	505627.071	6275	3172.80	4746.773
DAWM1	26.814	482868.649	482895.463	5168	2495.60	4194.590
DAWM2	30.460	459866.829	459897.289	5081	2336.73	4613.717
DAWM3	22.727	368506.408	368529.135	5081	1872.49	3915.173
DAWM4	28.364	389600.886	389629.249	5081	1979.70	4404.154
DAWM5	20.948	359891.180	359912.129	5081	1828.71	3565.901
DAWM6	21.186	338063.774	338084.960	5081	1717.81	3705.610
PAWM1	18.857	300213.272	300232.129	5081	1525.48	2990.434
PAWM2	21.087	294411.904	294432.991	5081	1496.01	3386.275

From this table, the following inferences can be drawn.

- From the 'Power' column, the total power of the proposed multipliers PAWM1 and PAWM2 is observed to be 300.21 μW and 294.41 μW respectively. This total power of the proposed multipliers is found to be lowest as compared to any other multipliers used for comparison. This power advantage can be attributed to the underlying architecture of proposed compressors.
- Considering the 'Delay' column, it is found that the delay of proposed multipliers is equal and the same. The value of this delay is equal to '5.081 ns'. Further, it is also observed that the delay of the proposed multipliers is found to be smaller and equal to that of DAWM2, DAWM3, DAWM4, DAWM5, and DAWM6.
- From the PDP column, it is found that the PAWM1 and PAWM2 are having a PDP metric of 1525.5 fJ and 1496.01 fJ respectively. These PDP values are found to lowest among all other multipliers considered for comparison. The lowest PDP can be attributed to the smaller power and delay of the proposed multipliers
- Again, from the area column, it is found that the area of the PAWM1 and PAWM2 is found to be 2990.43 μm^2 and 3386.27 μm^2, this advantage in the area can be attributed to the lower gate count of the proposed compressors.

Considering the aforementioned inferences, the proposed multipliers are found to be excellent in terms of overall DMs as compared to any other multiplier used for comparison. Thus, the proposed multipliers can be considered as the best candidature for image processing applications, where the energy and area efficient architectures are a paramount concern.

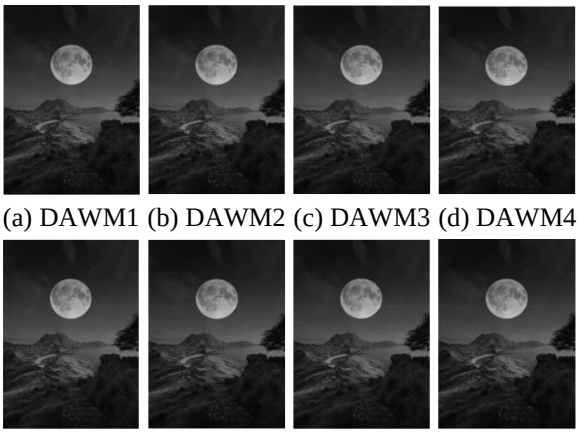
3.2. APPLICATION OF PROPOSED MULTIPLIER

This section illustrates the use of proposed multipliers in the context of image processing applications. Image multiplication is widely used in image processing applications such as image scaling, image sharpening, etc. [15]. In this paper, the proposed multipliers are evaluated for image multiplication. Here two test images are multiplied to produce a new image.

(a) (b) (c)

Fig.4. Multiplication of test images using 8×8 exact Wallace multipliers (a) Image-1 (b) Image-2 (c) Exact Multiplication

The test images selected for multiplication are i) Image 1 (Fig.4(a)) ii) Image 2 (Fig.4(b)). Both test images are of type 'portable network graphics' (png) having a size $m \times n$, with $m=n=512$. The reason for selecting these test images is that they have opposite features, as required for image analysis. The total number of pixels in each test image is 512×512=262144. Each pixel size is an 8-bit unsigned integer with a minimum value of '0' and a maximum value of '255'. A pixel by pixel multiplication is carried out to multiply two test images. The multiplication results of test images using the exact 8×8 multiplier is shown in Fig.4(c). To compare the multipliers in terms of PSNR, an image multiplication is extended on all other multiplier designs and their resultant images are shown in Fig.5.



(a) DAWM1 (b) DAWM2 (c) DAWM3 (d) DAWM4

(e) DAWM5 (f) DAWM6 (g) PAWM1 (h) PAWM2

Fig.5. Multiplication of test images using 8×8 designed and proposed approximate Wallace multipliers

To assess the resultant image quality, the 'peak signal-to-noise ratio' (PSNR) of all multipliers is computed with respect to Fig.4(c), using MATLAB tool. The calculated PSNR values for all designs have been tabulated in Table.7. From Table.7, the PSNR of PAWM1 and PAWM2 is found to be 51dB and 51.355dB respectively, which is comparable against other high PSNR multipliers under comparison. The PSNR values of these proposed multipliers are found to be better than DAWM2, DAWM4, and DAWM6 and comparable with DAWM1, DAWM3, and DAWM5. Thus, the proposed PAWM1 and PAWM2 are found to be a choice in terms of PSNR as compared to other multiplier designs.

Table.7. PSNR comparison of designed and proposed approximate Wallace multipliers

AWM	PSNR (dB)
DAWM1	52.996
DAWM2	50.688
DAWM3	52.106
DAWM4	48.744
DAWM5	55.511
DAWM6	47.386
PAWM1	51.000
PAWM2	51.355

4. CONCLUSION

This research paper, presented the design and analysis of two proposed unsigned 8×8 approximate Wallace multipliers: PAWM1 and PAWM2, using approximate compressors. The performance of these proposed multipliers has also been compared the other designed multipliers based on the reported approximate compressors. The comparison results show that the

proposed multipliers outperform against other multipliers in terms of DMs. Further from the image multiplication illustration, it is proved that the proposed multipliers are also capable of producing good quality images with the best and comparable PSNR. Thus, considering the overall inferences, the proposed multipliers are found to be a good and justified choice for image processing applications where the need for energy and area efficiency along with a good PSNR value are a paramount concern.

REFERENCES

[1] R. R. Osorio and G. Rodriguez, "Truncated SIMD Multiplier Architecture for Approximate Computing in Low-Power Programmable Processors", *IEEE Access*, Vol. 7, pp. 56353-56366, 2019.

[2] H. Jiang, C. Liu, F. Lombardi and J. Han, "Low-Power Approximate Unsigned Multipliers with Configurable Error Recovery", *IEEE Transactions on Circuits and Systems-I: Regular Papers*, Vol. 66, No. 1, pp. 189-202, 2019.

[3] L.B. Soares, M.M. Azevedo Da Rosa, C.M. Diniz, E.A.C. Costa and S. Bampi, "Design Methodology to Explore Hybrid, Approximate Adders for Energy-Efficient Image and Video Processing Accelerators", *IEEE Transactions on Circuits and Systems-I: Regular Papers*, Vol. 66, No. 6, pp. 2137-2150, 2019.

[4] I. Alouani, H. Ahangari, O. Ozturk and S. Nair, "A Novel Heterogeneous Approximate Multiplier for Low Power and High Performance", *IEEE Embedded System Letters*, Vol. 10, No. 2, pp. 45-48, 2018.

[5] S. Ataei and J.E. Stine, "A 64 kB Approximate SRAM Architecture for Low-Power Video Applications", *IEEE Embedded System Letters*, Vol. 10, No. 1, pp. 10-13, 2018.

[6] Minho Ha and Sunggu Lee, "Multipliers with Approximate 4:2 Compressors and Error Recovery Modules", *IEEE Embedded Systems Letters*, Vol. 10, No. 1, pp. 6-9, 2018.

[7] M. Ostal, A. Ibrahim, H. Chible and M. Valle, "Inexact Arithmetic Circuits for Energy Efficient IoT Sensors Data Processing", *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 1-4, 2018.

[8] W. Liu, J. Xu, D. Wang, C. Wang, P. Montuschi and F. Lombardi, "Design and Evaluation of Approximate Logarithmic Multipliers for Low Power Error-Tolerant Application", *IEEE Transactions on Circuits and Systems-I: Regular Papers*, Vol. 65, No. 9, pp. 2856-2868, 2018.

[9] C.V. Gowdar, M.C. Parameshwara and S Sonoli, "Comparative Analysis of Various Approximate Full Adders under RTL Codes", *ICTACT Journal on Microelectronics*, Vol. 6, No 2, pp. 947-952, 2020.

[10] C.V. Gowdar, M.C. Parameshwara and S Sonoli, "Approximate Full Adders for Multimedia Processing Applications", *Proceedings of IEEE International Conference for Innovation in Technology*, pp. 1-4, 2020.

[11] M.C. Parameshwara and H.C. Srinivasaiah, "Low-Power Hybrid 1-Bit Full Adder Circuit for Energy Efficient Arithmetic Applications", *Journal of Circuits, Systems, and Computers*, Vol. 26, No. 1, pp. 1-15, 2017.

[12] A. Dalloo, A. Najafi and A. Garcia-Ortiz, "Systematic Design of an Approximate Adder: The Optimized Lower Part Constant-OR Adder", *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, Vol. 26, No. 8, pp. 1595-1599, 2018.

[13] R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha and M. Pedram, "RoBA Multiplier: A Rounding-Based Approximate Multiplier for High-Speed yet Energy-Efficient Digital Signal Processing", *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, Vol. 25, No. 2, pp. 393-401, 2017.

[14] M.C. Parameshwara and H.C. Srinivasaiah, "Partial Product Compression Methods: A Study and Performance Comparison using a Tree Structured Multipliers", *International Journal of Engineering Research and General Science*, Vol. 4, No. 2, pp. 749-756, 2016.

[15] H.A.F. Almurib., T. Nandha Kumar, and F. Lombardi, "Inexact Designs for Approximate Low Power Addition by Cell Replacement", *Proceedings of IEEE International Conference on Design, Automation, and Test*, pp. 660-665, 2016.

[16] A. Momeni, J. Han, P. Montuschi and F. Lombardi, "Design and Analysis of Approximate Compressors for Multiplication", *IEEE Transactions on Computers*, Vol. 64, No. 4, pp. 984-994, 2015.

[17] Z. Yang, J. Han and F. Lombardi, "Approximate Compressors for Error-Resilient Multiplier Design", *Proceedings of IEEE International Conference on Defect and Fault Tolerance in VLSI and Nanotechnology Systems*, pp. 1-14, 2015.

[18] G. Vaibhav, M. Debabrata, R. Anand and R. Kaushik, "Low-Power Digital Signal Processing using Approximate Adders", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 32, No. 1, pp. 124-137, 2013.

[19] Z. Yang, A. Jain, J. Liang, J. Han and F. Lombardi, "Approximate XOR/XNOR-based Adders for Inexact Computing", *Proceedings of IEEE International Conference on Nanotechnology*, pp. 690-693, 2013.

[20] A.B. Kahng and S. Kang, "Accuracy-Configurable Adder for Approximate Arithmetic Designs", *Proceedings of IEEE International Conference on Design Auto*, pp. 820-825, 2012.

[21] D. Shin and S.K. Gupta, "Approximate Logic Synthesis for Error Tolerant Applications", *Proceedings of IEEE International Conference on Design, Automation, and Test*, pp. 1-4, 2010.

[22] 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 Transactions on Circuits and Systems-I: Regular Papers*, Vol. 57, No. 4, pp. 850-862, 2010.

[23] N. Zhu, W.L. Goh, W. Zhang, K.S. Yeo and Z.H. Kong, "Design of Low-Power High Speed Truncation-Error-Tolerant Adder and Its Application in Digital Signal Processing", *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, Vol. 18, No. 8, pp. 1225-1229, 2010.