An efficient look up table based approximate adder for field programmable gate array

Hadise Ramezani, Majid Mohammadi, Amir Sabbagh Molahosseini
Department of Computer Engineering, Kerman Branch, Islamic Azad University, Kerman, Iran

Article Info

Article history:

Received Jun 25, 2021
Revised Oct 28, 2021
Accepted Nov 26, 2021

Keywords:

Adders
Approximate computing
FPGA
Gaussian smoothing filter
LUTs

ABSTRACT

The approximate computing is an alternative computing approach which can lead to high-performance implementation of audio and image processing as well as deep learning applications. However, most of the available approximate adders have been designed using application specific integrated circuits (ASICs), and they would not result in an efficient implementation on field programmable gate arrays (FPGAs). In this paper, we have designed a new approximate adder customized for efficient implementation on FPGAs, and then it has been used to build the Gaussian filter. The experimental results of the implementation of Gaussian filter based on the proposed approximate adder on a Virtex-7 FPGA, indicated that the resource utilization has decreased by 20-51%, and the designed filter delay based on the modified design methodology for building approximate adders for FPGA-based systems (MDeMAS) adder has improved 10-35%, due to the obtained output quality.

Corresponding Author:

Majid Mohammadi
Department of Computer Engineering, Kerman Branch, Islamic Azad University
Kerman, Iran
Email: mohammadi@uk.ac.ir

1. INTRODUCTION

Extensive applications of small-scale embedded systems to high-performance computational systems can be implemented by using field programmable gate arrays (FPGAs). The advantages of FPGA are short time to market, high flexibility, and run-time configurability [1]. However, FPGA systems consume more power or energy compared to application specific integrated circuits (ASICs), despite the availability of hardware accelerators and special co-processors [2]. Hence, new FPGA-based energy efficiency optimization methods should also be developed and employed in addition to using conventional power reduction techniques. One of these new approaches is the approximate computations, which can simultaneously provide high performance and energy efficiency [3].

Approximate computations deal with the accuracy of intermediate or final computations in contrast to the delay, area, and power or energy consumption. This type of trade-off is very advantageous in applications, which are inherently resistant to faults [4]. Applications that are resistant to fault produce acceptable output, despite the reduced accuracy of computations. All applications such as image or video processing, data mining, and machine learning are resistant to fault, and therefore approximate computations can be used for them [5]. Approximate computational methods can be applied at different levels of computation, which extend from logic gates to compilers and programming languages [6]. A lot of software and hardware researches have been performed in the field of approximate computation [7]-[12]. Voltage over-scaling [13], [14] and functional approximation [15] are two main categories in hardware approximate computations.

Software approximate computational methods are divided into two main categories: i) loop perforation and function approximation [16], [17], and ii) programming language support [18], [19]. Most of the hardware approximate computation procedures focus on the basic computation modules like adders [20], [21] and multipliers [22], [23]. The full adder circuit is a basic unit in digital arithmetic and logic circuits [24].

Many studies have been carried out on approximate computations. most of them have been designed and implemented as ASIC. Since, there are structurally differences between ASIC and FPGA, the approximate computations approaches designed for ASIC cannot be designed and implemented directly for FPGA and achieve the same results of ASIC [25]. In this study, we intend to investigate approximate computations with respect to FPGA architecture, that the carry speculative adder (CSPA) and design methodology for building approximate adders for FPGA-based systems (DeMAS) adders have been used in its design. In section 2 we introduce existing approximate adders. In section 3, the proposed approximate adder will introduce. In section 4, implementation tests on proposed adder and comparison with other approximate adders will explored, and in section 5 we conclude our paper.

2. EXISTING APPROXIMATE ADDERS

2.1. Carry speculative adder (CSPA)

A CSPA is indicated in Figure 1 [26]. This adder is similar to the carry save adder (CSA), with the difference that the CSPA includes one unit of sum generator, two units of internal carry generators (generating carries with 0 input, and generating carries with 1 input) and one unit of carry predictor in each block. The output of carry predictor of 1th block is used to choose the output of one of the two units of internal carry generator of (i+1)th block for the sum generator unit. In the carry predictor unit, $k<x$ bit is only used, that k is bits of each block and x denotes bits that from each block used for predicting carry.

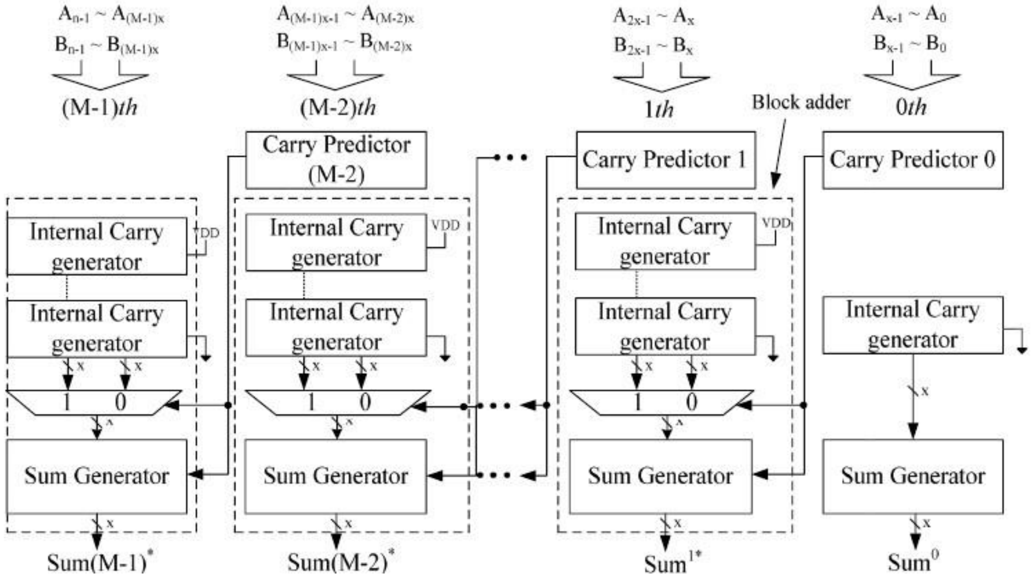

Figure 1. Structure of the CSPA adder [26]

2.2. DeMAS adder

Prabakaran *et al.* [27] introduces eight look up table (LUT)-based approximate adders called DeMAS for the Xilinx Virtex-7 Series FPGA, and the architectural features of the target FPGA device has been considered in their design. These adders are shown in Figure 2 (note that the adder-6 is 2 bits, and also has the highest accuracy among others). In this adder, the output value of truth table for LUT3 is "8E" and the output value of truth table for LUT5 is "E080FEF8". In this method, for designing an *N*-bit adder where these approximate adders are used for *k* least-significant bits (LSBs), and accurate adders are used for the (*N*-*k*) most significant bits (MSBs).

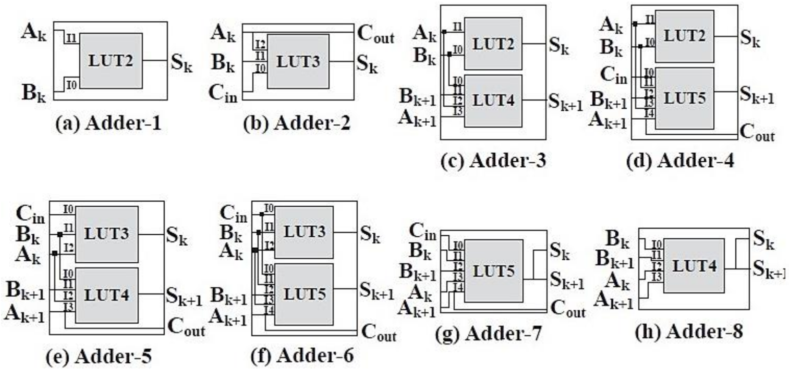

Figure 2. LUT-based approximate adders (a) adder-1 (b) adder-2 (c) adder-3 (d) adder-4 (e) adder-5 (f) adder-6 (g) adder-7 and (h) adder-8 [27]

3. METHODOLOGY

The architecture of the proposed adder is shown in Figure 3. In this architecture, the adder is divided into x-bit blocks similar to CSPA. A carry predictor is used in each block, but instead of the gates related to internal adder of each block, a DeMAS adder renamed to MDeMAS has been used. The truth table indicates a precise 2-bit adder, DeMAS Adder-6 and the proposed adder in Figure 4. In the DeMAS collector, the high value bit of the first digit, A1, is considered as Cout. The high value bit of the sum, S_1, is generated by a LUT5 with inputs A1A0B1B0C0 and output value "E080FEF8", and the low value bit, S_0, is generated by a LUT3 with input A0B0C0 and output value "8E".

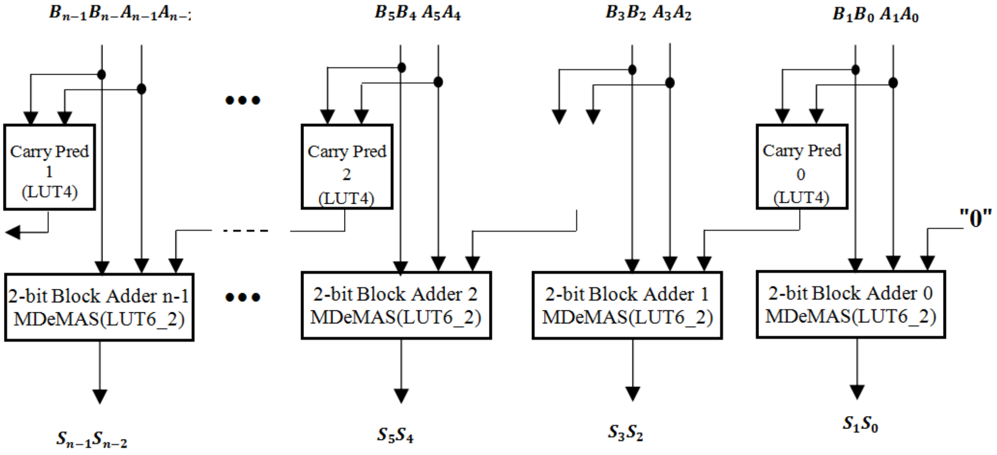
$B_{n-1}B_{n-}A_{n-1}A_{n-}$:
$B_5B_4\ A_5A_4$
$B_3B_2\ A_3A_2$
$B_1B_0\ A_1A_0$
Carry Pred 1 (LUT4)
Carry Pred 2 (LUT4)
Carry Pred 0 (LUT4)
"0"
2-bit Block Adder n-1 MDeMAS(LUT6_2)
2-bit Block Adder 2 MDeMAS(LUT6_2)
2-bit Block Adder 1 MDeMAS(LUT6_2)
2-bit Block Adder 0 MDeMAS(LUT6_2)
$S_{n-1}S_{n-2}$
S_5S_4
S_3S_2
S_1S_0

Figure 1. Architecture of the suggested adder using 2-bites blocks

As shown in Table 1, in the DeMAS method for Adder-6, the highest magnitude of error is 2, the sum of magnitudes of errors is 28 and the error average is 0.875, which is relatively high. In the proposed adder, the following modifications have been made to DeMAS to reduce the approximation error: The carry predictor presented in [26] has been used for generating the Cout. The inputs of carry predictor are A1A0B1B0, and can be implemented with one LUT4. In the DeMAS method, the output carry is generated directly based on the input of A1, whereas just one carry has been used in the proposed method, therefore the accuracy has been increased.

In the proposed method, we have generated a 2-bit sum using one LUT6-2. We generate the output of sum according to the generated carry by the carry predictor, so that there is a slight difference between the values of output sum and the precise sum. As illustrated in Table 1, the predicted output carry differs from the precise output carry in four states. In the cases where the predicted carry is equal to the exact output carry, we make the approximate sum equal to the sum, and in non-equal states, we set the approximate sum based on the carry, so that there is the least difference with the z precise sum.

Table 1. Truth table of exact, approximate DeMAS and proposed approximate MDeMAS adder

					Accurate Adder			DeMAS Ader-6				Proposed Adder (MDeMAS)			
A1	A0	B1	B0	Cln	Cout	S1	S0	Cout	S1	S0	Error magnitude	Cout	S1	S0	Error magnitude
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0
0	0	0	1	1	0	1	0	0	1	1	1	0	1	0	0
0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0
0	0	1	0	1	0	1	1	0	1	0	1	0	1	1	0
0	0	1	1	0	0	1	1	0	1	0	1	0	1	1	0
0	0	1	1	1	1	0	0	0	1	1	1	0	1	1	1
0	1	0	0	0	0	0	1	0	0	0	1	0	0	1	0
0	1	0	0	1	0	1	0	0	1	1	1	0	1	0	0
0	1	0	1	0	0	1	0	0	1	1	1	0	1	0	0
0	1	0	1	1	0	1	1	0	1	1	0	0	1	1	0
0	1	1	0	0	0	1	1	0	1	0	1	0	1	1	0
0	1	1	0	1	1	0	0	0	1	1	1	0	1	1	1
0	1	1	1	0	1	0	0	0	1	1	1	1	0	0	0
0	1	1	1	1	1	0	1	0	1	1	2	1	0	1	0
1	0	0	0	0	0	1	0	1	0	0	2	0	1	0	0
1	0	0	0	1	0	1	1	1	0	0	1	0	1	1	0
1	0	0	1	0	0	1	1	1	0	0	1	0	1	1	0
1	0	0	1	1	1	0	0	1	0	1	1	0	1	1	1
1	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0
1	0	1	0	1	1	0	1	1	0	0	1	1	0	1	0
1	0	1	1	0	1	0	1	1	0	0	1	1	0	1	0
1	0	1	1	1	1	1	0	1	1	1	1	1	1	0	0
1	1	0	0	0	0	1	1	1	0	0	1	0	1	1	0
1	1	0	0	1	1	0	0	1	0	1	1	0	1	1	1
1	1	0	1	0	1	0	0	1	0	1	1	1	0	0	0
1	1	0	1	1	1	0	1	1	0	1	0	1	0	1	0
1	1	1	0	0	1	0	1	1	0	0	1	1	0	1	0
1	1	1	0	1	1	1	0	1	1	1	1	1	1	0	0
1	1	1	1	0	1	1	0	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0

In the DeMAS method, k/2 of DeMAS adder-6 and a precise (N-k)-bit adder have been used. they generate the N-bit adder. This adder is shown in Figure 4. The main problem with this N-bit adder is the presence of the carry chain between blocks of the DeMAS adder-6 and inside the precise adder, which increases the delay. In the proposed adder, the following alterations were made to the DeMAS multi-bit adder to reduce the delay.

As shown in Figure 4, we have replaced the DeMAS adder-6 with a carry predictor block and an MDeMAS adder, and removed the carry chain between blocks. Since the accuracy of proposed MDeMAS adder and predictor carry is higher than the DeMAS adder-6, the removal of carry chain does not reduce overall accuracy of the N-bit adder. In the N-bit DeMAS adder, a precise adder is employed to compute the high value (N-k)-bit of sum, which increases the total delay of the gate. In the proposed adder, we have removed this precise adder and increased the accuracy of generation of carry and the MDeMAS internal adder of each block.

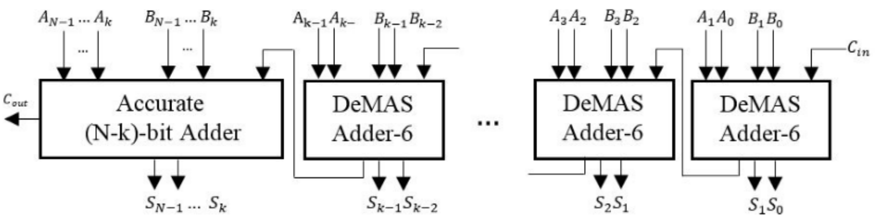

Figure 2. Truth table of exact, approximate DeMAS and proposed approximate MDeMAS adder

4. RESULTS AND DISCUSSION

4.1. Experimental setup

To evaluate the proposed adder and approximate adders, we have used them in a 2D 3×3 Gaussian convolution filter. The architecture of this filter is indicated in Figure 5, which is used to blur the images. The employed Gaussian kernel has the mean distribution of 0 and σ=1. The CSPA [26], DeMAS [27] and the proposed MDeMAS adders have been designed and implemented using the very high-speed integrated circuit (VHSIC) hardware description language (VHDL) language and in the Xilinx ISE environment. Using each of these adders, we have designed a Gaussian filter based on architecture [28] and implemented and synthesized them to evaluate and compare the area and delay on the Virtex-7 family device 7VX330T. MATLAB has been used to analyze the accuracy and quality of the designed filter outputs.

4.2. Accuracy result

The output quality of Gaussian filters in which different approximate adders are used is shown by the peak signal to noise ratio (PSNR) and structural index similarity (SSIM) criteria in Table 2. The Lena image with 512×512 dimensions has been used as the filter input. Approximate adders with different configurations have been employed in the Gaussian filter. The value of BlockSize represents the size of blocks in CSPA and MDeMAS adders. The MDeMAS adder is made with only the size of block 2 for optimal use of FPGA Virtex-7 resources. The value of approxBits in the DeMAS adder represents the number of low value bits of sum obtained by approximation. In the filter made based on DeMAS, the high approxBits the lower the output quality, but the delay will improve. The output quality of the filter made on the basis of the proposed MDeMAS adder possesses an accuracy equivalent to the filter made on the basis of the DeMAS with only 2-bit approximation.

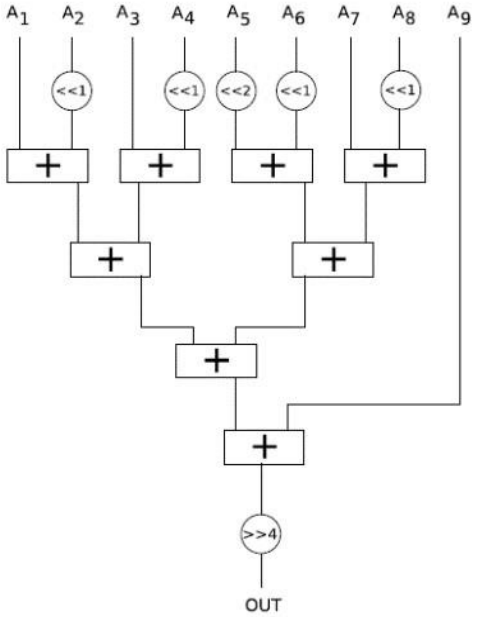

Figure 5. Gaussian filter convolution [28]

Table 2. Results of Gaussian filter circuits quality based on different approximate adders

Gaussian filter		PSNR	SSIM
2DGSF-CSPA	BlockSize=2	21.45	0.44
	BlockSize=4	22.33	0.47
2DGSF-DeMAS	approxBits=2	22.96	0.45
	approxBits=4	21.85	0.42
	approxBits=6	18.88	0.36
2DGSF-MDeMAS	BlockSize=2	22.79	0.44

4.3. Delay and area results

The CSPA, DeMAS adders and the proposed adder have been used for making the Gaussian filter shown in Figure 5. the designed filters have been synthesized on Virtex-7 family device 7VX330T. Results have been listed in Table 3. As determined from the results, the gate of designed filter has the lowest area by using the approximate MDeMAS adder. Its delay is in the range of a filter, which is based on the DeMAS with approximate number of 6 bits with the lowest output quality. The input pixels of filters are grey, and therefore the input values of adders are 8-bit.

Table 3. Synthesis results of constructed Gaussian filter based on different approximate adders

Gaussian filter		Area (LUT)	Delay(ns)	Area×Delay
2DGSF-CSPA	BlockSize=2	61	4.59	279.99
	BlockSize=4	61	4.992	304.512
2DGSF-DeMAS	approxBits=2	112	7.613	852.656
	approxBits=4	96	6.017	577.632
	approxBits=6	82	4.421	362.522
2DGSF-MDeMAS	BlockSize=2	51	4.89	249.39

As shown in Figure 6, the delay of the filter made with the approximate DeMAS adder with a 6-bit approximation compared with filter delays designed with our proposed adder is less. This is because DeMAS uses 6 bits of approximation, therefore produces an inappropriate image quality. According to the results of the output quality, in Gaussian filter design based on the proposed method, we can simultaneously reduce both the delay and the output quality, and reduce the utilization of resources considerably as shown in Figure 7.

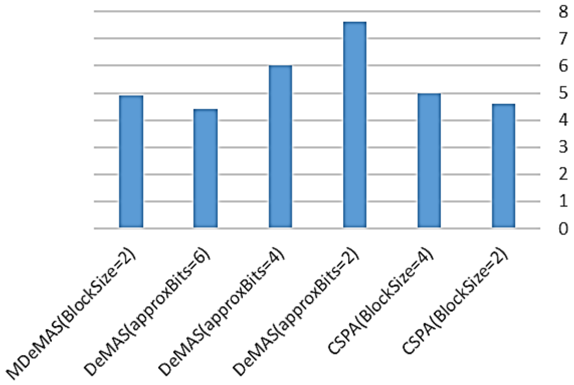

Figure 6. Bar chart of the delay of the constructed filters based on different approximate adders

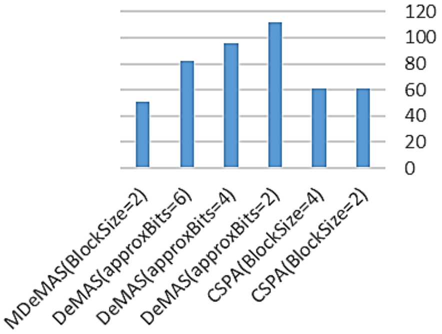

Figure 7. Bar chart of comparison of the consumed amount of constructed Gaussian filters using different approximate adders

Product of area and delay is a measure that must have a trade-off, since to reduce the delay, we need more area. Therefore, for optimizing these two measures, product of them must be minimum. Figure 8, shows the proposed method is superior to other methods in terms of this criterion. We could simultaneously reduce the delay and output quality. Moreover, reduce the rate of utilization of resources, considerably. According to the results of output quality of Table 2 and the results of synthesis of Table 3 in the Gaussian filter designed based on the proposed adder,we have such reductions.

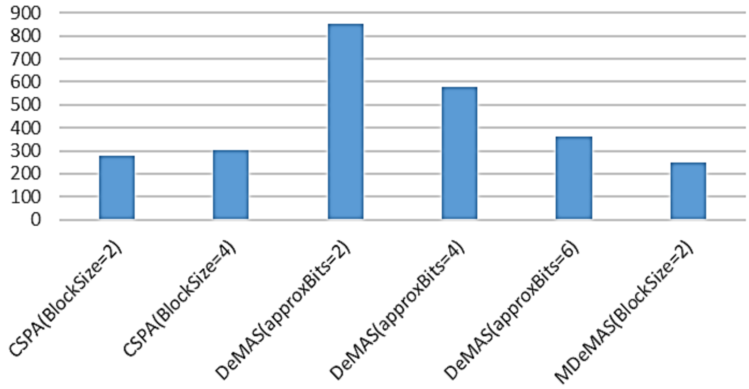

Figure 8. Bar chart of product of delay and consumed amount of constructed Gaussian filters of the constructed filters using different approximate adders

5. CONCLUSION

In this study, we have designed a new approximate adder to implement on the FPGA, that the CSPA and DeMAS adders have been used in its design. The proposed approximate adder was used to make the Gaussian filter and then implemented on the FPGA Virtex-7. The results demonstrated that the resource utilization has decreased by 20-51%, and the delay of designed filter based on MDeMAS adder has been improved by 10-35% due to the obtained output quality.

REFERENCES

[1] A. DeHon and J. Wawrzynek, "Reconfigurable computing: what, why, and implications for design automation," *Proceedings 1999 Design* Automation *Conference (Cat. No. 99CH36361)*, 1999, pp. 610-615, doi: 10.1109/DAC.1999.782016.

[2] F. Krach, B. Frackelton, J. Carletta and R. Veillette, "FPGA-based implementation of digital control for a magnetic bearing," *Proceedings of the 2003 American Control Conference*, 2003, pp. 1080-1085, doi: 10.1109/ACC.2003.1239730.

[3] S. Mittal, "A survey of techniques for approximate computing," *ACM Computing Surveys (CSUR)*, vol. 48, no. 4, pp. 1-33, 2016, doi: 10.1145/2893356.

[4] V. K. Chippa, S. T. Chakradhar, K. Roy, and A. Raghunathan, "Analysis and characterization of inherent application resilience for approximate computing," in *Proceedings of the 50th Annual Design Automation Conference*, 2013, pp. 1-9, doi: 10.1145/2463209.2488873.

[5] C. Chang, A. S. Molahosseini, A. A. E. Zarandi, and T. F. Tay, "Residue Number Systems: A New Paradigm to Datapath Optimization for Low-Power and High-Performance Digital Signal Processing Applications," in *IEEE Circuits and Systems Magazine*, vol. 15, no. 4, pp. 26-44, Fourthquarter, 2015, doi: 10.1109/MCAS.2015.2484118.

[6] M. Shafique, R. Hafiz, S. Rehman, W. El-Harouni, and J. Henkel, "Cross-layer approximate computing: From logic to architectures," in *Proceedings of the 53rd Annual Design Automation Conference*, 2016, pp. 1-6, doi: 10.1145/2897937.2906199.

[7] H. Ramezani, M. Mohammadi, and A. S. Molahoseini, "An Efficient Implementation of Low-Latency Two-Dimensional Gaussian Smoothing Filter using Approximate Carry-Save Adder," *Journal of Circuits, Systems and Computers*, vol. 30, no. 02, pp. 2150021, 2021, doi: 10.1142/S0218126621500213.

[8] N. Zhu, W. L. Goh, and K. S. Yeo, "Ultra low-power high-speed flexible Probabilistic Adder for Error-Tolerant Applications," *2011 International SoC Design Conference*, 2011, pp. 393-396, doi: 10.1109/ISOCC.2011.6138614.

[9] 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," in *IEEE Transactions on Very Large Scale Integration (VLSI*) Systems, vol. 18, no. 8, pp. 1225-1229, Aug. 2010, doi: 10.1109/TVLSI.2009.2020591.

[10] N. Zhu, W. L. Goh, G. Wang, and K. S. Yeo, "Enhanced low-power high-speed adder for error-tolerant application," 2*010 International SoC Design Conference*, 2010, pp. 323-327, doi: 10.1109/SOCDC.2010.5682905.

[11] R. Ye, T. Wang, F. Yuan, R. Kumar, and Q. Xu, "On reconfiguration-oriented approximate adder design and its application," *2013 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2013, pp. 48-54, doi: 10.1109/ICCAD.2013.6691096.

[12] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, "A low latency generic accuracy configurable adder," *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2015, pp. 1-6, doi: 10.1145/2744769.2744778.

[13] K. V. Palem, L. N. Chakrapani, Z. M. Kedem, A. Lingamneni, and K. K. Muntimadugu, "Sustaining moore's law in embedded computing through probabilistic and approximate design: retrospects and prospects," in *Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems*, 2009, pp. 1-10, doi: 10.1145/1629395.1629397.

[14] M. Imani, A. Rahimi, and T. S. Rosing, "Resistive configurable associative memory for approximate computing," *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2016, pp. 1327-1332.

[15] D. Shin and S. K. Gupta, "Approximate logic synthesis for error tolerant applications," *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)*, 2010, pp. 957-960, doi: 10.1109/DATE.2010.5456913.

[16] W. Baek and T. M. Chilimbi, "Green: A framework for supporting energy-conscious programming using controlled approximation," in *Proceedings of the 31st ACM SIGPLAN Conference on Programming Language Design and Implementation*, 2010, pp. 198-209, doi: 10.1145/1806596.1806620.

[17] S. Misailovic, M. Carbin, S. Achour, Z. Qi, and M. C. Rinard, "Chisel: Reliability-and accuracy-aware optimization of approximate computational kernels," *ACM Sigplan Notices*, vol. 49, no. 10, pp. 309-328, 2014, doi: 10.1145/2660193.2660231.

[18] H. Esmaeilzadeh, A. Sampson, L. Ceze, and D. Burger, "Architecture support for disciplined approximate programming," in *Proceedings of the seventeenth international conference on Architectural Support for Programming Languages and Operating Systems*, 2012, pp. 301-312, doi: 10.1145/2248487.2151008.

[19] D. Mahajan *et al.*, "Axilog: Abstractions for Approximate Hardware Design and Reuse," in *IEEE Micro*, vol. 35, no. 5, pp. 16-30, Sept.-Oct. 2015, doi: 10.1109/MM.2015.108.

[20] M. A. Hanif, R. Hafiz, O. Hasan, and M. Shafique, "QuAd: Design and analysis of quality-area optimal low-latency approximate adders," in *Proceedings of the 54th Annual Design Automation Conferenc*, 2017, pp. 1-6, doi: 10.1145/3061639.3062306.

[21] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-Power Digital Signal Processing Using Approximate Adders," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 1, pp. 124-137, Jan. 2013, doi: 10.1109/TCAD.2012.2217962.

[22] S. Rehman, W. El-Harouni, M. Shafique, A. Kumar, J. Henkel, and J. Henkel, "Architectural-space exploration of approximate multipliers," *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2016, pp. 1-8, doi: 10.1145/2966986.2967005.

[23] P. Kulkarni, P. Gupta, and M. Ercegovac, "Trading Accuracy for Power with an Underdesigned Multiplier Architecture," *2011 24th Internatioal Conference on VLSI Design*, 2011, pp. 346-351, doi: 10.1109/VLSID.2011.51.

[24] M. Mohammadi, M. Mohammadi, and S. Gorgin, "An efficient design of full adder in quantum-dot cellular automata (QCA) technology," *Microelectronics Journal*, vol. 50, pp. 35-43, 2016, doi: 10.1016/j.mejo.2016.02.004.

[25] A. S. Molahosseini, L. S. De Sousa, and C.-H. Chang, "Embedded systems design with special arithmetic and number systems,". Springer, 2017, doi: 10.1007/978-3-319-49742-6.

[26] C. Lin, Y.-M. Yang, and C.-C. Lin, "High-performance low-power carry speculative addition with variable latency," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 9, pp. 1591-1603, 2014. doi: 10.1109/TVLSI.2014.2355217.

[27] B. S. Prabakaran *et al.*, "DeMAS: An efficient design methodology for building approximate adders for FPGA-based systems," *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2018, pp. 917-920, doi: 10.23919/DATE.2018.8342140.

[28] S.-A. Li, C.-C Wong, C.-Y Yang, and L.-F Cheng, "Hardware acceleration for image processing," *Conference Towards Autonomous Robotic Systems*, 2012, pp. 436–437, doi: 10.1007/978-3-642-32527-4_48.

BIOGRAPHIES OF AUTHORS

Hadise Ramezani Received a bachelor's degree from Shiraz university, and M.Sc. degree in kerman Islamic Azad University in the field of computer engineering. She is pursuing the Ph.D. degree in the field of Computer Systems Architecture in Kerman Islamic Azad university. Her research interests are approximate computing and image processing. She can be contacted at email: h.ramezani@iauk.ac.ir.

Majid Mohammadi Was born in 16 September of 1966. He got B.Sc. of Electronic engineering in 1989 in Khajenasir university; M.Sc. of Control engineering in 1994 in Tehran university, and Ph.D. of Computer Architecture in 2009 in Shahid Beheshti university. He is lecturer in computer engineering department of Shahid Bahonar university since 1996. He interested in Analysis and design of quantum circuits, Reversible logic circuir design, multiple valued logic, Digital signal processing and its applications in sound, watermarking, steganography and cryptography. He can be contacted at email: mohammadi@uk.ac.ir.

Amir Sabbagh Molahosseini Received the B.Sc. degree from the Shahid Bahonar University of Kerman, Iran, in 2005, and the M.Sc. and Ph.D. degrees (Hons.) in computer engineering from Islamic Azad University, Science and Research Branch, Tehran, Iran, in 2007 and 2010, respectively. He was a Visiting Researcher with the Signal Processing Systems Group, Instituto de Engenharia de Sistemas e Computadores, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal. He is currently an Assistant Professor with the Department of Computer Engineering, Islamic Azad University Kerman Branch, Kerman, Iran, and leads the High-Performance Computer Arithmetic Group, since 2010. His current research interests are computer arithmetic with special emphasis on residue number systems, and alternative computing systems with special emphasis on approximate computing. He can be contacted at email: sabbagh@iauk.ac.ir.