A 64-Bit Rounding Based Approximate Multiplier for High Speed Digital Signal Processing

[1]Asmita Dhote, [2]Dr. Vikas Gupta
[1]M.Tech scholar, [2]Professor & Head
[1&2]Department of Electronics and Communication,
[1&2]Technocrats Institute of Technology, Bhopal, India.

Abstract : A multiplier is an electronic circuit used in digital electronics, such as a computer, to multiply two binary numbers. It is built using binary adders. A variety of computer arithmetic techniques can be used to implement a digital multiplier. Multipliers play a key role in the present digital signal handling and different applications. A multiplier is a combinational logic circuit that we use to multiply binary digits with advances in innovation, numerous scientists have attempted and are endeavoring to structure multipliers which offer both of the following plan targets – high speed, low power utilization, consistency of design and henceforth less zone or even mix of them in one multiplier in this way making them appropriate for different high speed, low power and minimal VLSI usage.

IndexTerms **– ROBA, VLSI, Multiplier, Speed, power, Area, Rounding.**

I. INTRODUCTION

Rounding technique is one of the most efficient methods for packing the input data before processing. This method has a potential to improve the circuit characteristics such as power and energy consumption, speed and area which is suitable method for the approximate computing. Approximate computing works very well to most of error resilient applications in the field of computer vision, image processing, pattern recognition, signal processing, scientific computing, and machine learning. Over past decade, research on these areas has given lots of opportunities in research. A multiplier is a fundamental block of computation and one of the most resource-consuming operations rounding input data requires major responsibility in maintaining the accuracy. With a basic intuition, it can be stated that, rounding lower bits results in less error compared to rounding higher bits. Thus, the proposed algorithm has assigned rounding weights with respect to the bit position value.
the execution of the multiplier can be incredibly improved. Be that as it may, the expenses are an unpredictable 'multiplexer' with zero, multiplicand, and twice multiplicand contributions, just as the carry-in information and one's supplement calculation required for negative numbers. Higher radix Stall's recoding can be utilized to additionally diminish the quantity of cycles however requires a significantly progressively complex multiplexer. Note that most iterative multipliers based on MBR neglect to successfully misuse the operand structure; accordingly, they are fixed cycle multipliers.

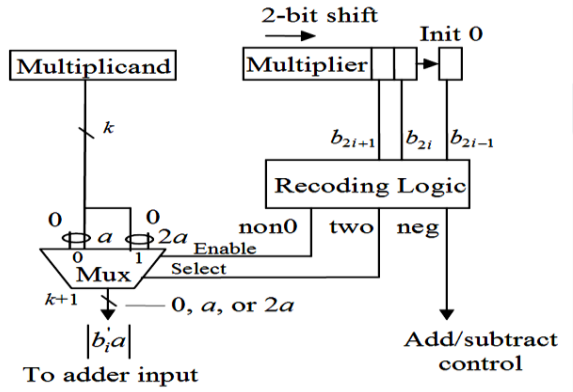

Figure 1: An iterative multiplier structure based radix-4 MBR**Error! Reference source not found.**

Notwithstanding the three foremost execution upgrade strategies listed above, there are extra procedures accessible for improving the execution of an iterative multiplier by diminishing the inertness per cycle and by planning effective structures for performing quick expansion, including, e.g., carry-look-ahead and carry select hardware.

If a positional numeral system is used, a natural way of multiplying numbers is taught in schools as long multiplication, sometimes called grade-school multiplication, sometimes called Standard Algorithm: multiply the multiplicand by each digit of the multiplier and then add up all the properly shifted results. It requires memorization of the multiplication table for single digits.

This is the usual algorithm for multiplying larger numbers by hand in base 10. Computers initially used a very similar shift and add algorithm in base 2, but modern processors have optimized circuitry for fast multiplications using more efficient algorithms, at the price of a more complex hardware realization. A person doing long multiplication on paper will write down all the products and then add them together; an abacus-user will sum the products as soon as each one is computed.

II. BACKGROUND

R. Zendegani et al., [1] propose three equipment executions of the approximate multiplier that incorporates one for the unsigned and two for the marked activities. The effectiveness of the proposed multiplier is assessed by contrasting its execution and those

of some approximate and precise multipliers utilizing diverse plan parameters. Moreover, the adequacy of the proposed approximate multiplier is considered in two picture handling applications, i.e., picture honing and smoothing.

S. Vahdat et al., [2] proposed approximate multiplier with a mean outright relative mistake in the scope of 11%-0.3% improves postponement, zone, and vitality utilization up to 41%, 90%, and 98%, individually, contrasted with those of the precise multiplier. It additionally outflanks other approximate multipliers as far as speed, zone, and vitality utilization. The proposed approximate multiplier has a practically Gaussian mistake appropriation with a close to zero mean esteem. We abuse it in the structure of a JPEG encoder, honing, and grouping applications. The outcomes demonstrate that the quality corruption of the yield is insignificant. Moreover, we recommend an exactness configurable TOSAM where the vitality utilization of the increase activity can be balanced based on the base required precision.

T. Su et al., [3] The technique comprises of three fundamental advances: 1) decide the weights (twofold encoding) of the yield bits; 2) remake the truncated multiplier utilizing useful blending and re-union; and 3) build the polynomial mark of the subsequent circuit. The technique has been tried on multipliers up to 256 bits with three truncation plans: Cancellation, D-truncation, and Truncation with Rounding. Exploratory outcomes are contrasted and the best in class SAT, SMT, and PC logarithmic solvers.

R. Saxena et al., [4] presents the minimization of the transistors required in designing the circuits and to reduce the power consumption of the circuits, the authors have referred some techniques to overcome these problems in this paper. By reviewing all these techniques, the authors try to implement the GDI technique to reduce the power consumption and transistors count or the area required to design the circuits.

P. Lohray et al., [5] proposed in this work. Reproduction results for three chose advancements show noteworthy enhancement for the circuit attributes as far as power, region, speed, and vitality for proposed multiplier in examination with their partners. Info information rounding design and the likelihood of the redundancy for adjusted qualities has been acquainted as two basic things with control the dimension of the exactness for each scope of the information with least expense on the equipment.

A. Ferozpuri et al.,[6] This work exhibits a high-speed FPGA usage for the NIST Cycle 1 PQC accommodation of Rainbow. We examine a high-speed structure that utilizes a parameterized framework solver, which can unravel a n-byn framework in n clock cycles. Contrasted with the past best in class, we decrease the quantity of required multipliers by practically half, speed up execution, and actualize Rainbow for higher security levels. Our plan underpins numerous parameter sets, which require activities in the fields GF(16) and GF(256). Furthermore, so as to make benchmarking simpler and more pleasant, our plan follows an all inclusive PQC equipment Programming interface, which allows for reasonable examination with other post-quantum signature plans. This plan is being made open-source to build straightforwardness and speed up further enhancement.

E. Hosseini et al., [7] In this work, another high-speed and low power unsigned increase structure is proposed: based on the proposed algorithm, the info bits of multiplier are broken into a few littler gatherings of bits and the augmentation of them are determined simultaneously. The last result of increase is created after a few rounds of the little gathering's outcomes collection. A 32*32-piece multiplier as indicated by the proposed structure is planned in 0.18um CMOS process. The general deferral of 32*32-piece multiplier is incredibly low and is just 2.1ns. The power utilization is 41mW.

I. Hatai et al., [8] This work presents a computationally effective equipment design for reconfigurable various consistent duplication square, which capacities as indicated by the accepted marked digit (CSD)- based vertical and flat basic sub-articulation disposal (VHCSE) algorithm. In the proposed design, the CSD decoded coefficient alongside 4-b normal sub-articulations (CSs) in the vertical bearing and 4-and 8-b CSs the even way decreases the required number of full adder cells and the adder profundities. This strategy helps in lessening zone utilization by diminishing the quantity of coefficient multiplier adders by 59% than that of the parallel VHCSE (VHBCSE) algorithm.

R. DiCecco et al., [9] We utilize these centers with our motor to prepare systems to show that an example width of 6 and mantissa width of 5 accomplishes exactness tantamount to single-accuracy coasting point for the MNIST and CIFAR-10 datasets. These outcomes are accomplished utilizing round-to-zero for the CPFP multipliers and round-to-closest for the CPFP adders, allowing for LUT reserve funds of 32.6% for the multipliers and 21.7% for the adders when contrasted with half-exactness drifting point, while utilizing a similar number of DSPs.

T. Su et al., [10] The technique comprises of three essential advances: 1) decide the weights (double encoding) of the yield bits; 2) recreate the truncated multiplier utilizing practical blending and re-union; and 3) develop the polynomial mark of the subsequent circuit. The technique has been tried on multipliers up to 256 bits with three truncation plans: Cancellation, D-truncation, and Truncation with Rounding. Exploratory outcomes are contrasted and the best in class SAT, SMT, and PC logarithmic solvers.

III. PROPOSED WORK

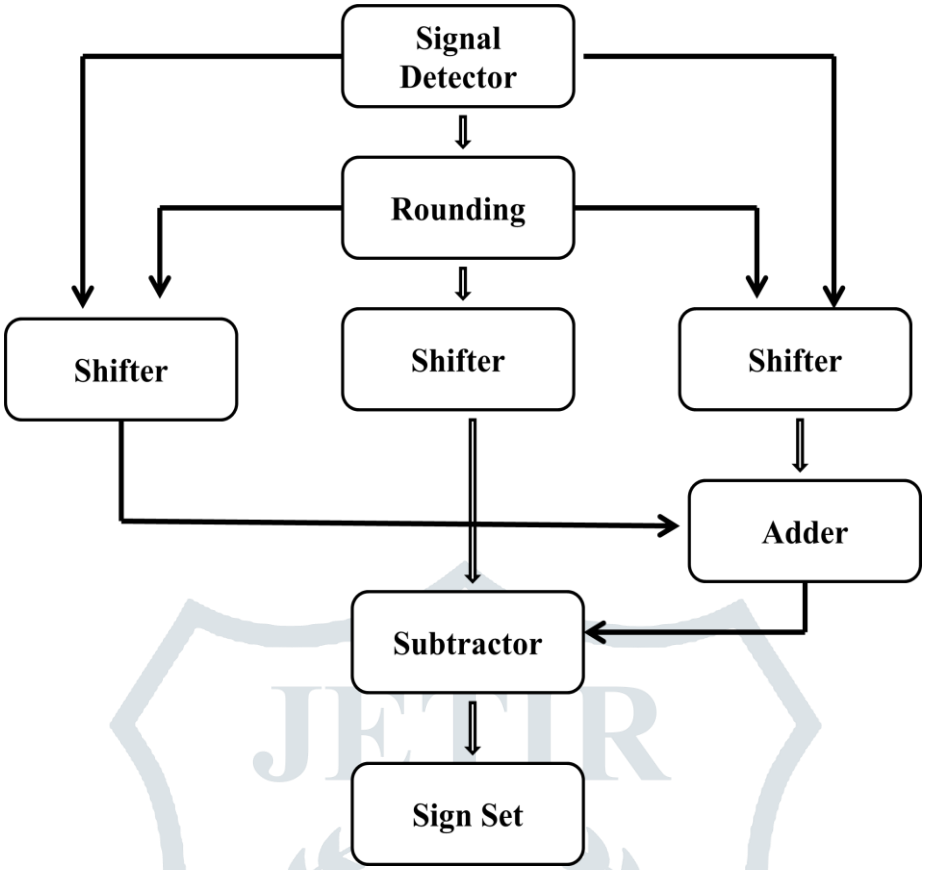

Figure 2: Flow Chart

Algorithm-

- In this work it is proposed to design and analyze the performance of the ROBA multiplier for high speed digital signal processing.
- Check different parameters like speed, Look up table, time etc.
- To design ROBA multiplier.
- Simulate and synthesis using Xilinx 14.7.
- To test with different input combination and check speed and accuracy.

An approximate multiplier that is high speed yet vitality proficient. The methodology is to round the operands to the closest type of two. Along these lines the computational serious piece of the augmentation is precluded improving speed and vitality utilization at the cost of a little blunder. The proposed methodology is relevant to both marked and unsigned augmentations. We propose three equipment executions of the approximate multiplier that incorporates one for the unsigned and two for the marked activities. The effectiveness of the proposed multiplier is assessed by contrasting its execution and those of some approximate and exact multipliers utilizing diverse plan parameters.

Multiplication Algorithm of ROBA Multiplier

The principle thought behind the proposed approximate multiplier is to utilize the simplicity of activity when the numbers are two to the power n (2n). To expand on the activity of the approximate multiplier, first, let us signify the adjusted quantities of the contribution of An and B by Ar and Br , individually.
The multiplication of A by B may be rewritten as

A X B = (Ar -A) X (Br - B) + Ar X B + Br X A – Ar X Br

The key perception is that the duplications of Ar × Br , Ar×B, and Br×A might be executed just by the move task. The equipment execution of (Ar − A)×(Br − B), be that as it may, is somewhat intricate. The weight of this term in the last outcome, which relies upon contrasts of the accurate numbers from their adjusted ones, is normally little. Henceforth, we propose to exclude this part from (1), streamlining the increase task. Henceforth, to play out the augmentation procedure.

IV. SIMULATION RESULT

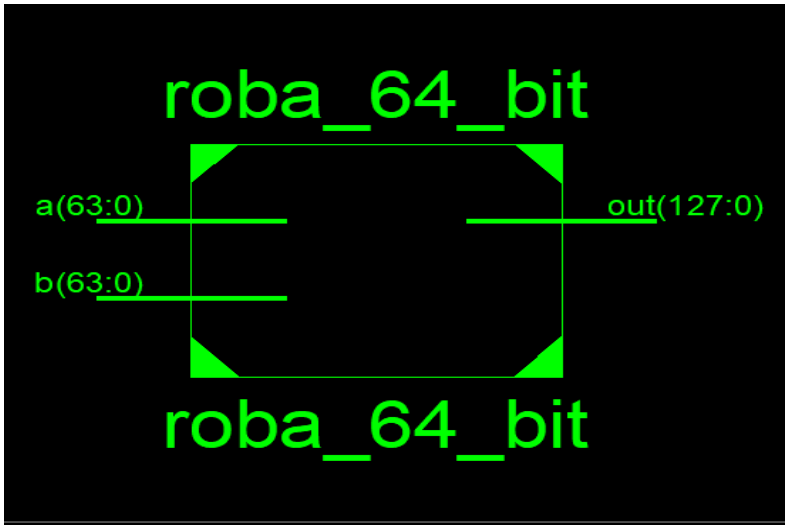

Figure 3: ROBA Top level module

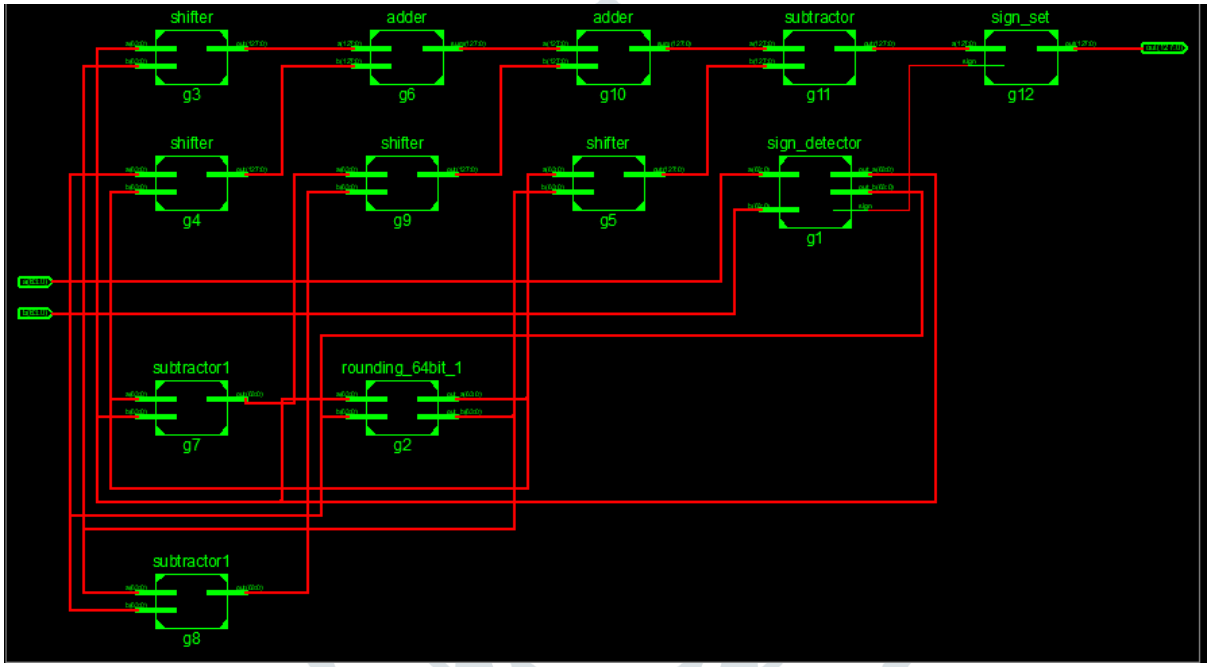

Figure 4: RTL of ROBA Multiplier

RTL view of proposed ROBA multiplier is showing in figure 4, all digital circuits of using component are showing clearly like adder, shifter, Subtractor sign set etc. This multiplier can be used for 8 bit, 16 bit and 32 bit and 64 bit processing.

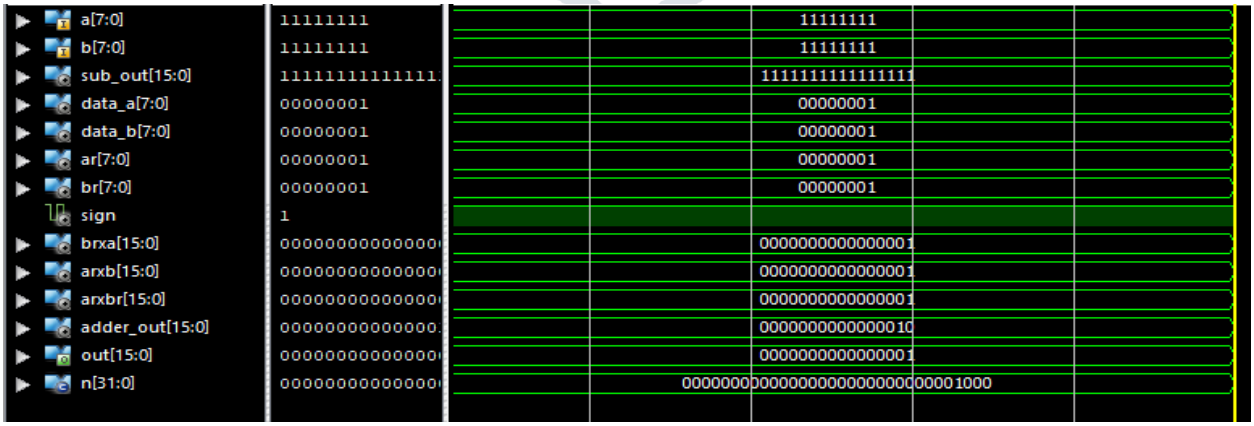

Figure 5: ROBA multiplier test bench

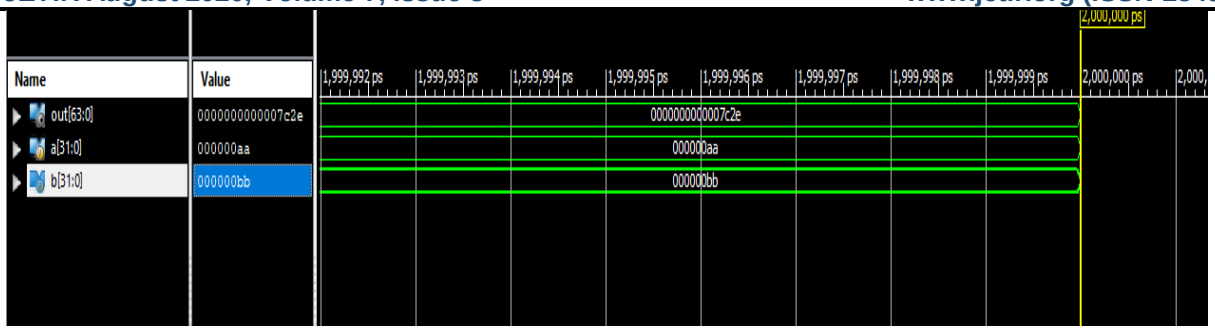

Figure 6: showing input and output in generated test bench

Figure 6 is showing the input and output parameter value. The input a is taken as 'aa' and input b is taken as 'bb' and using proposed ROBA multiplier the output C is 7C2E.

Table 1: Simulation Parameter and Comparison with previous work

Sr No.	Parameters	Previous work	Proposed work
1	Type of Multiplier	ROBA -32 bit	ROBA – 64 bit
2	Area	13.31%	12.25%
3	Delay	21.79ns	42.800ns
4	Accuracy rate	90 %	95%
6	Power	1.03mW	0.42mW
7	PDP (Power delay product)	22.44	17.97

Table 1 showing comparison of proposed work with previous work, so it can be seen that proposed multiplier gives better result than existing work.

V. CONCLUSION

The quantity of include/subtract activities is additionally decreased using sidestep strategies. By and large, 66.7% of the incomplete item age tasks are supplanted with a straightforward detour to the moving structure and carry spread is completely dispensed with also. In this way, the multifaceted nature of the equipment usage is drastically decreased when contrasted with ordinary strategies, including changed Stall recoding and contending ROBA systems. This methodology accomplishes a general speed-up just as diminished power utilization which is especially basic in versatile interactive media applications. Therefore in this paper, design and analysis of rounding based approximate multiplier for digital signal processing. Consequently obviously such different is skilled to give quick increase of digital signal with high exactness. It additionally requires less investment and expends less territory. Presently, ROBA multiplier can be utilized in various digital signal applications. .

REFERENCES

1. 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," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 2, pp. 393-401, Feb. 2017.
2. S. Vahdat, M. Kamal, A. Afzali-Kusha and M. Pedram, "TOSAM: An Energy-Efficient Truncation- and Rounding-Based Scalable Approximate Multiplier," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*.
3. T. Su, C. Yu, A. Yasin and M. Ciesielski, "Formal Verification of Truncated Multipliers Using Algebraic Approach and Re-Synthesis," *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Bochum, 2017, pp. 415-420
4. R. Saxena and K. Sharma, "A Comparative Review on ALU using CMOS and GDI techniques for Power Dissipation and Propagation Delay", IJOSTHE, vol. 7, no. 1, p. 4, Feb. 2020. https://doi.org/10.24113/ojssports.v7i1.119
5. P. Lohray, S. Gali, S. Rangisetti and T. Nikoubin, "Rounding Technique Analysis for Power-Area & Energy Efficient Approximate Multiplier Design," *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2019, pp. 0420-0425.
6. A. Ferozpuri and K. Gaj, "High-speed FPGA Implementation of the NIST Round 1 Rainbow Signature Scheme," *2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, Cancun, Mexico, 2018, pp. 1-8.
7. E. Hosseini, M. Mousazadeh and A. Amini, "High-Speed 32*32 bit Multiplier in 0.18um CMOS Process," *2018 25th International Conference "Mixed Design of Integrated Circuits and System" (MIXDES)*, Gdynia, 2018, pp. 154-159.
8. I. Hatai, I. Chakrabarti and S. Banerjee, "A Computationally Efficient Reconfigurable Constant Multiplication Architecture Based on CSD Decoded Vertical–Horizontal Common Sub-Expression Elimination Algorithm," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 1, pp. 130-140, Jan. 2018.
9. R. DiCecco, L. Sun and P. Chow, "FPGA-based training of convolutional neural networks with a reduced precision floating-point library," *2017 International Conference on Field Programmable Technology (ICFPT)*, Melbourne, VIC, 2017, pp. 239-242.

10. T. Su, C. Yu, A. Yasin and M. Ciesielski, "Formal Verification of Truncated Multipliers Using Algebraic Approach and Re-Synthesis," *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Bochum, 2017, pp. 415-420.
11. A. Alavian and M. C. Rotkowitz, "Improving ADMM-based optimization of Mixed Integer objectives," *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, 2017, pp. 1-6.
12. D. De Caro, E. Napoli, D. Esposito, G. Castellano, N. Petra and A. G. M. Strollo, "Minimizing Coefficients Wordlength for Piecewise-Polynomial Hardware Function Evaluation With Exact or Faithful Rounding," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 64, no. 5, pp. 1187-1200, May 2017.
13. Mang Liao and A. Chakrabortty, "A Round-Robin ADMM algorithm for identifying data-manipulators in power system estimation," *2016 American Control Conference (ACC)*, Boston, MA, 2016, pp. 3539-3544.
14. J. Hormigo and J. Villalba, "Measuring Improvement When Using HUB Formats to Implement Floating-Point Systems Under Round-to-Nearest," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2369-2377, June 2016.