A Verilog Implementation of 64-bit Rounding Based Approximate Multiplier Design

[1]Shipransi Shrivastava, [2]Prof. Ashish Raghuwanshi

[1]M.Tech Scholar, [2]Assistant Professor
[1&2]Department of Electronics and Communication,
[1&2]IES College of Technology, Bhopal (M.P.), India.

Abstract : A multiplier is an electronic circuit used in digital electronics, such as a computer, to multiply two binary or any numbers. It is built using binary adders, shifter etc. A variety of computer arithmetic techniques can be used to implement a digital multiplier. In this paper proposed 64 bit Rounding Based Approximate Multiplier design. Proposed design gives better performance in terms of area, delay, power than existing.

***IndexTerms* - ROBA, VLSI, Multiplier, Delay, Area, Power.**

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.

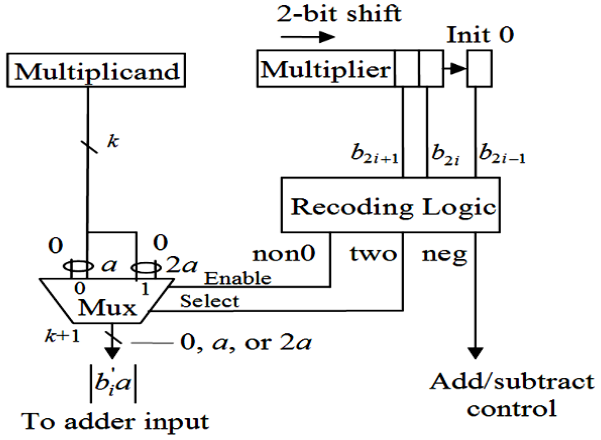


Figure 1: An iterative multiplier structure based radix-4 MBR**Error! Reference source not found.**

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.

II. LITERATURE SURVEY

R. Zendegani et al., [1] In this work, it is propose an approximate multiplier that is high speed yet vitality productive. The methodology is to round the operands to the closest example of two. Along these lines the computational serious piece of the augmentation is overlooked improving speed and vitality utilization at the cost of a little mistake. The proposed methodology is appropriate to both marked and unsigned duplications. 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 precise multipliers utilizing diverse plan parameters.

S. Vahdat et al., [2] An adaptable approximate multiplier, called truncation-and rounding-based versatile approximate multiplier (TOSAM) is introduced, which decreases the quantity of fractional items by truncating every one of the info operands based on their driving one-piece position. In the proposed plan, duplication is performed by move, include, and little fixed-width augmentation tasks bringing about extensive enhancements in the vitality utilization and territory occupation contrasted with those of the precise multiplier. To improve the complete precision, input operands of the duplication part are adjusted to the closest odd number. Since information operands are truncated based on their driving one-piece positions, the exactness turns out to be feebly reliant on the width of the information operands and the multiplier ends up adaptable. Higher enhancements in structure parameters (e.g., territory and vitality utilization) can be accomplished as the info operand widths increment.

T. Su et al., [3] This work introduces a formal way to deal with check multipliers that approximate whole number increase by yield truncation. The strategy is based on separating polynomial mark of a truncated multiplier utilizing logarithmic revamping. To proficiently process the polynomial mark, a multiplier reproduction approach is utilized to build the exact multi-plier from the truncated one. 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.

M. J. Schulte et al., [4] This work presents equipment plans that produce precisely adjusted outcomes for the elements of complementary, square-root, 2/sup x/, and log/sub 2/(x). These plans utilize polynomial guess in which the terms in the estimate are produced in parallel, and afterward summed by utilizing a multi-operand adder. To decrease the quantity of terms in the estimation, the information interim is parceled into subintervals of equivalent size, and distinctive coefficients are utilized for each subinterval. The coefficients utilized in the estimate are at first decided based on the Chebyshev arrangement guess. They are then changed in accordance with acquire precisely adjusted outcomes for all sources of info. Equipment plans are introduced, and deferral and territory correlations are made based on the level of the approximating polynomial and the exactness of the last outcome.

P. Lohray et al., [5] Approximate figuring is one of most appropriate productive information preparing for blunder strong applications, for example, signal and picture handling, PC vision, AI, information mining and so on. Approximate registering lessens exactness which is worthy as an expense of expanding the circuit attributes relies upon the application. Attractive exactness is the edge point for controlling the exchange off, among precision and circuit attributes under the control of the circuit architect. In this work, the rounding procedure is presented as a proficient strategy for controlling this exchange off. In such manner multiplier circuits as a basic structure obstruct for registering in the vast majority of the processors have been considered for the assessment of the rounding system productivity. The effect of the rounding strategy is investigated by examination of circuit qualities for three multipliers. These three multipliers are the traditional Wallace tree precise multiplier, DRUM [4] the as of late proposed approximate multiplier and the adjusted based approximate multiplier 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.

III. SIMULATION AND RESULT

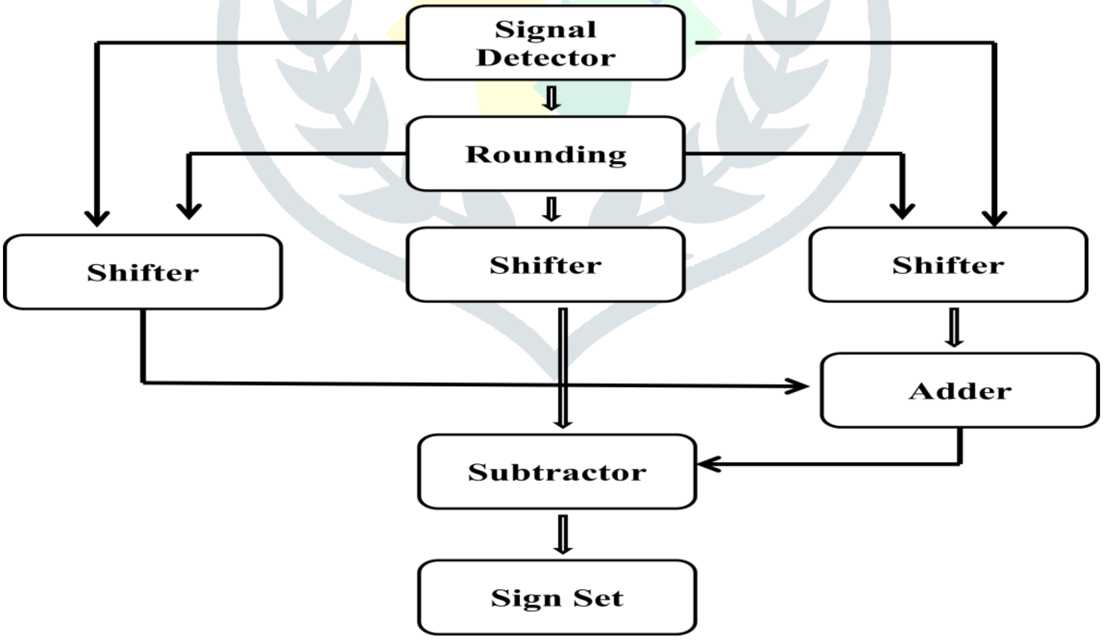


Figure 2: Flow Chart

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.

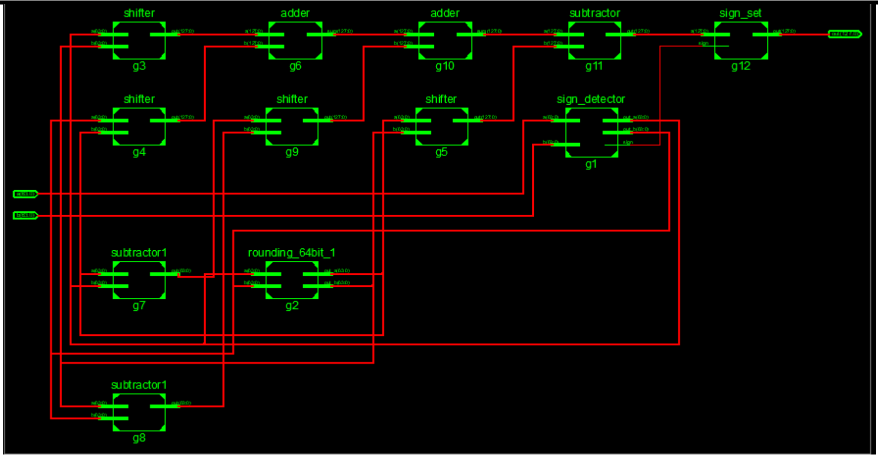


Figure 3: RTL of ROBA Multiplier

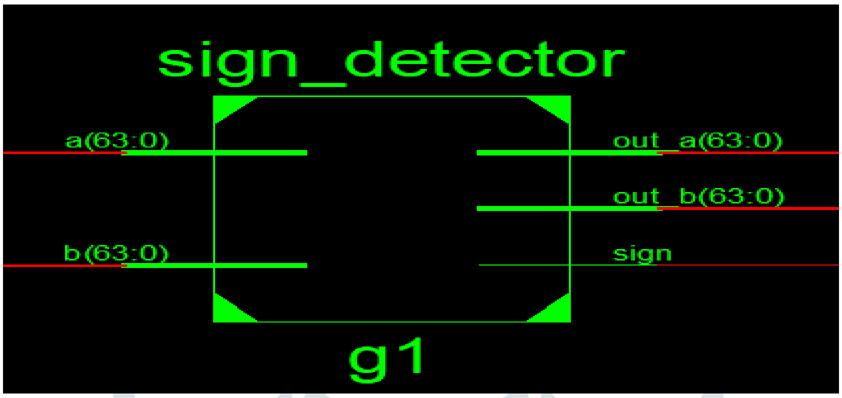


Figure 4: ROBA sign detector

In figure 4, showing one component of proposed multiplier i.e shifter, which can shift input data and send for next process.

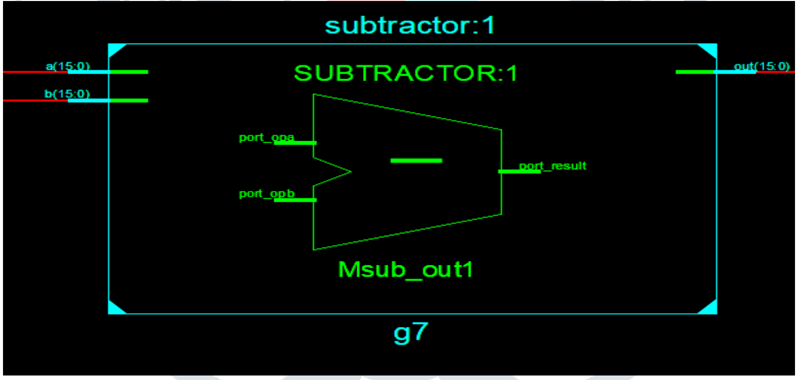
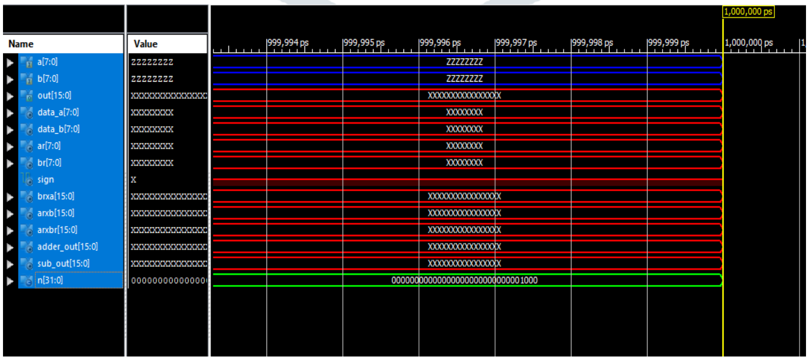


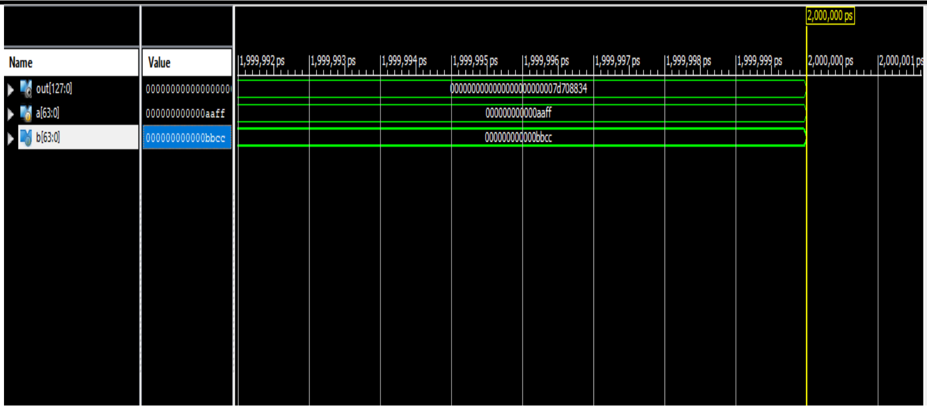
Figure 5: ROBA Subtractor



Name	Value
a[7:0]	zzzzzzzz
b[7:0]	zzzzzzzz
out[15:0]	xxxxxxxxxxxxxxxx
data_a[7:0]	xxxxxxxx
data_b[7:0]	xxxxxxxx
ar[7:0]	xxxxxxxx
br[7:0]	xxxxxxxx
sign	x
brxa[15:0]	xxxxxxxxxxxxxxxx
arxb[15:0]	xxxxxxxxxxxxxxxx
arxbr[15:0]	xxxxxxxxxxxxxxxx
adder_out[15:0]	xxxxxxxxxxxxxxxx
sub_out[15:0]	xxxxxxxxxxxxxxxx
n[31:0]	00000000000000000000000000001000

Figure 6: High impendence test bench bar

In figure 6, showing test bench bar for all possible value, which is also known as high impendence.



Name	Value
out[127:0]	0000000000000000000000007d708834
a[63:0]	000000000000aaff
b[63:0]	000000000000bbcc

Figure 7: 64 Bit ROBA multiplier test bench in binary number

In figure 7, showing input a is aaff and input b is bbcc and output is 7D708834

Table 1: Comparison with Previous and proposed 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

Therefore design and synthesis of ROBA multiplier using Xilinx verilog and find proposed multiplier is better than previous multiplier.

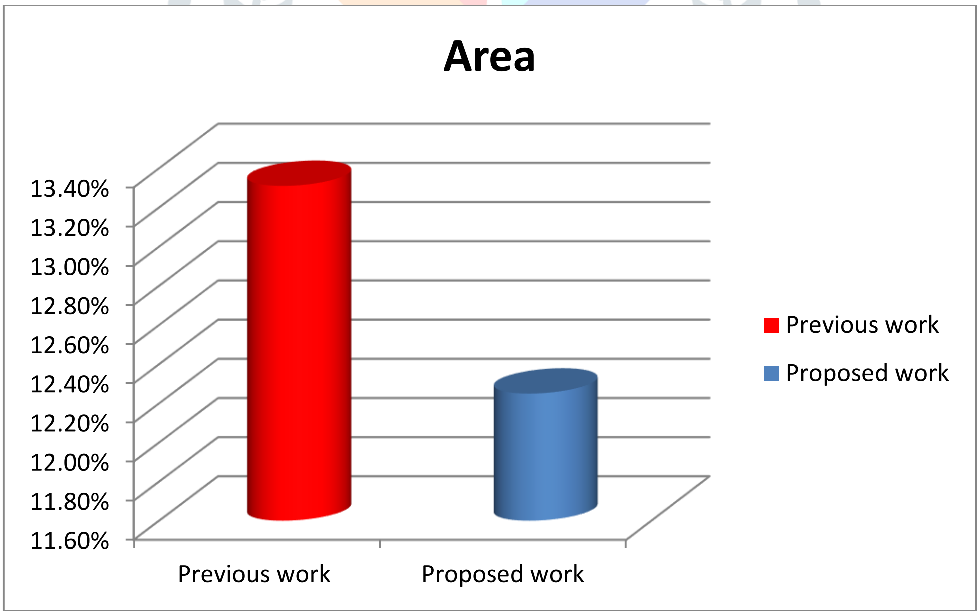


Figure 8: Area

In figure 8, showing area of proposed work and previous work. This is graphical representation of result.

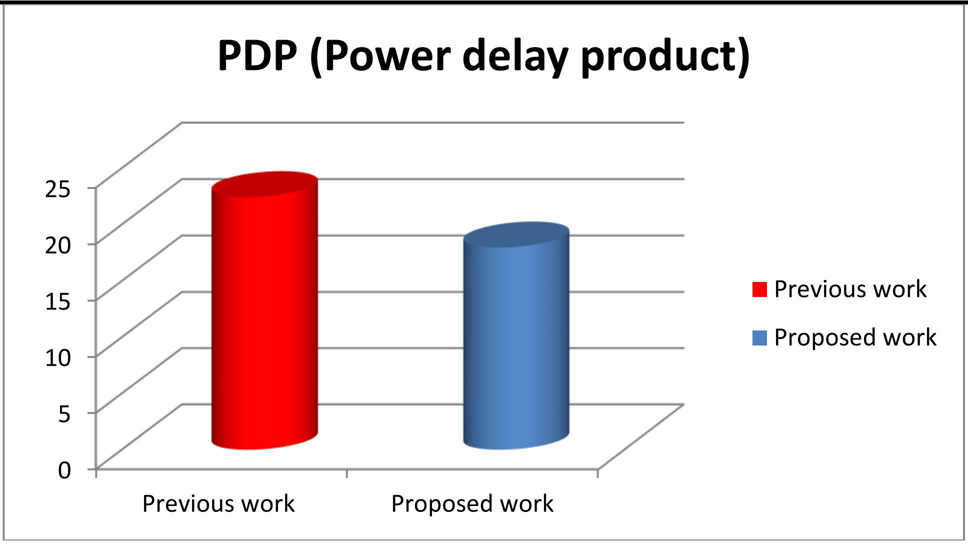


Figure 9: Power delay product

In figure 9, showing PDP of proposed work and previous work. This is graphical representation of result and it is clear that proposed method can be calculate fast sothat overall system speed will be improved.

IV. CONCLUSION

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.

In future work,

- Modified ROBA multiplier which can give more accurate multiplication.
- Real time multiplication using different digital signal application.
- Make hardware implementation using FPGA kit.

REFERENCE

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. M. J. Schulte and E. E. Swartzlander, "Hardware designs for exactly rounded elementary functions," in *IEEE Transactions on Computers*, vol. 43, no. 8, pp. 964-973, Aug. 1994.
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.
15. E. G. Walters, "24-bit significand multiplier for FPGA floating-point multiplication," *2015 49th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, 2015, pp. 717-721.