



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

¹Aamir Siddiqui, ²Dr. Uday Panwar

¹M.Tech Scholar, ²Associate Professor

Department of Electronics and Communication Engineering,
Sagar Institute of Research & 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. 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. This paper proposed proposed 64 bit ROBA multiplier with improved performance then previous. Previous work, designed only 16 bit and 32 bit ROBA multiplier. Proposed 64-bit ROBA multiplier reduced 10% area than existing 32-bit multiplier, save 60% power. Proposed multiplier is giving 5% more accuracy than existing. Delay is significant constant so it can be say that it is improved than previous. The simulation is performed Xilinx 14.7 software.

IndexTerms - ROBA, VLSI, Multiplier, Speed, Power, Xilinx, Area.

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.

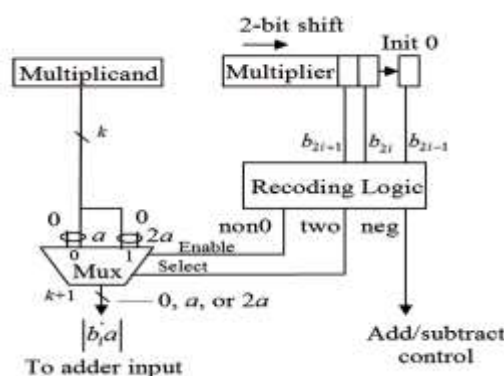


Figure 1: An iterative multiplier structure based radix-4 MBR

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.

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.

The multiplication algorithm for an N bit multiplicand by N bit multiplier is shown below:

$Y = Y_{n-1} Y_{n-2} \dots Y_2 Y_1 Y_0$ Multiplicand

$X = X_{n-1} X_{n-2} \dots X_2 X_1 X_0$ Multiplier

Also, gates are utilized to create the Fractional Items, PP, On the off chance that the multiplicand is N-bits and the Multiplier is M-bits, at that point there is $N * M$ halfway item. How the fractional items are produced or summed up is the distinction between the diverse models of different multipliers.

Conversely, run of the mill iterative multipliers use a couple of equipment practical units over and over to create incomplete items consecutively and add each recently created item to those recently collected. The fundamental qualities of iterative multipliers are little territory utilization; diminished stick checks and wire length, and high clock rate. In addition, by executing various emphases that are information subordinate, the vitality effectiveness can be enormously improved with respect to cluster and parallel multipliers. Here, vitality proficiency alludes to the vitality required per task, e.g., nano-Joules/operation. Thusly, the decision between actualizing a parallel or cluster multiplier rather than an iterative multiplier in some random structure is commonly an exchange off of computational speed against zone necessity and vitality productivity. In this work, we present another iterative multiplier plan that gives basic structure, high throughput, and high vitality productivity, making it especially reasonable for organization in low power and low region applications.

II. BACKGROUND

P. Lohray et al., [1] proposed approximate multiplier and the rounded based approximate multiplier proposed in this work. Simulation results for three selected technologies show significant improvement on the circuit characteristics in terms of power, area, speed, and energy for proposed multiplier in comparison with their counterparts. Input data rounding pattern and the probability of the repetition for rounded values has been introduced as two essential items to control the level of the accuracy for each range of the data with minimum cost on the hardware.

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.

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. Ferozpur 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-by-n 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.

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 CFPF multipliers and round-to-closest for the CFPF 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.

A. Alavian et al., [11] gives a class of blended whole number projects where the issue is curved aside from a vector of discrete factors. Two techniques dependent on the Substituting Heading Strategy for Multipliers (ADMM) are introduced. The primary, which has showed up in the new writing, copies the discrete variable, with one duplicate permitted to shift constantly. These outcomes in a straightforward projection, or adjusting, to decide the discrete variable at every emphasis.

D. De Caro et al., [12] gives whole number Direct Programming (ILP), upgrades the polynomial coefficients considering all mistake segments at the same time. This gives two points of interest. Initially, we can acquire precisely adjusted approximations; furthermore, for reliably adjusted interpolators, we evade any overdesign because of critical suspicions on blunder segments, advancing in this way the subsequent equipment.

Mang Liao et al., [13] propose an elective calculation where the focal boss, rather than figuring the normal, utilizes a Cooperative strategy to produce the agreement variable, and show that by following the advancement of this agreement variable it is conceivable to recognize which assessor is malevolent. We examine the union properties of this adjusted ADMM calculation, and represent its viability utilizing reproduction results.

J. Hormigo et al., [14] the usage of these units, in light of essential designs, shows that the Center configurations all the while improve territory, speed, and force utilization. Moreover, in light of the information acquired from the blend, a Center single-accuracy viper is ~14% quicker yet burns-through 38% less zone and 26% less force than the ordinary snake. Additionally, a Center point single-accuracy multiplier is 17% quicker, utilizes 22% less territory, and devours somewhat less force than the regular multiplier.

III. METHODOLOGY

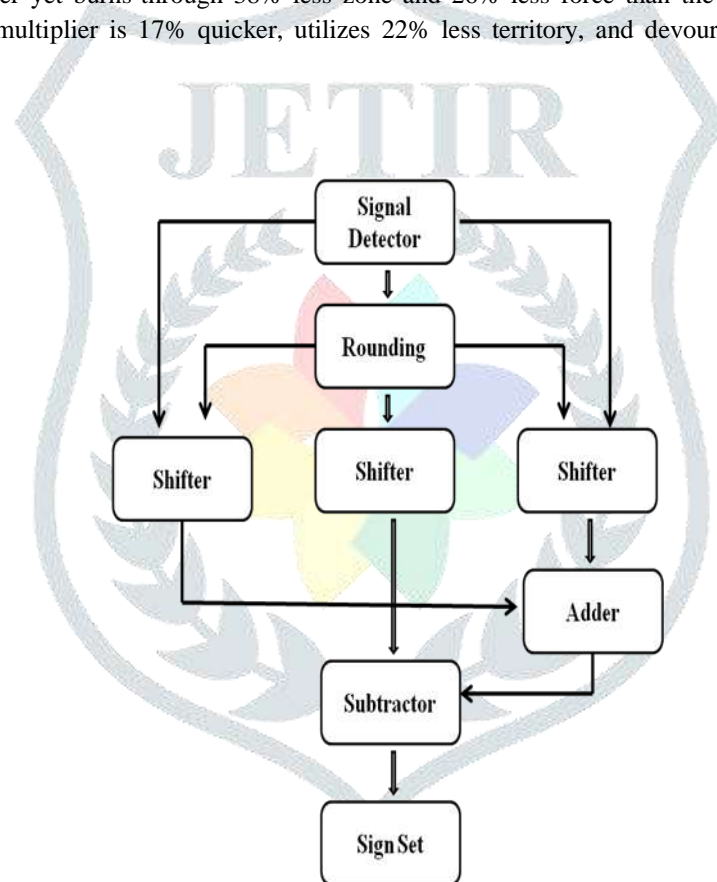


Figure 2: Flow Chart

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.

The fundamental thought behind the inexact multiplier is to make utilization of the simplicity of activity when the numbers are two to the power n ($2n$). To expound on the task of the inexact multiplier, first, let us mean the adjusted quantities of the contribution of A and B by A_r and B_r , separately. The increase of A by B might be changed as

$$A \times B = (A_r - A) \times (B_r - B) + A_r \times B + B_r \times A - A_r \times B_r \quad (1)$$

The key perception is that the duplications of $A_r \times B_r$, $A_r \times B$, and $B_r \times A$ might be executed just by the move task. The equipment execution of $(A_r - A) \times (B_r - B)$, be that as it may, is fairly perplexing. The heaviness of this term in the last outcome, which relies upon contrasts of the precise numbers from their adjusted ones, is commonly little. Subsequently, we propose to exclude this part from (1), streamlining the augmentation activity. Consequently, to play out the duplication procedure, the accompanying articulation is utilized:

$$A \times B \approx A_r \times B + B_r \times A - A_r \times B_r \quad (2)$$

Subsequently, one can play out the increase activity utilizing three move and two expansion/subtraction tasks. In this methodology, the closest qualities for A and B as $2n$ ought to be resolved. At the point when the estimation of A (or B) is equivalent to the $3 \times 2^{p-2}$ (where p is a self-assertive positive number bigger than one), it has two closest qualities as $2n$ with

equivalent outright contrasts that are $2p$ and $2p-1$. While the two qualities lead to a similar impact on the exactness of the multiplier, choosing the bigger one (aside from the instance of $p = 2$) prompts a littler equipment execution for deciding the closest adjusted esteem, and thus, it is considered in this paper. It begins from the way that the numbers as $3 \times 2p-2$ are considered as couldn't care less in both gathering together and down improving the procedure, and littler rationale articulations might be accomplished in the event that they are utilized in the gathering together. The main special case is for three, which for this situation; two is considered as its closest incentive in the surmised multiplier

IV. SIMULATION AND RESULTS

The simulation is performed using Xilinx ISE 14.7 software.

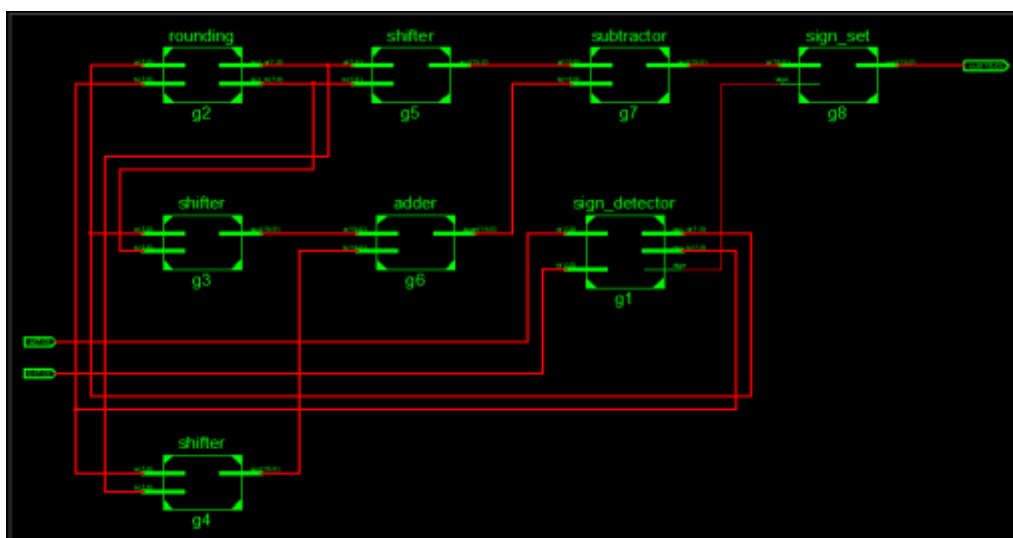


Figure 3: RTL of ROBA Multiplier

RTL view of proposed ROBA multiplier is showing in figure 3, 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.

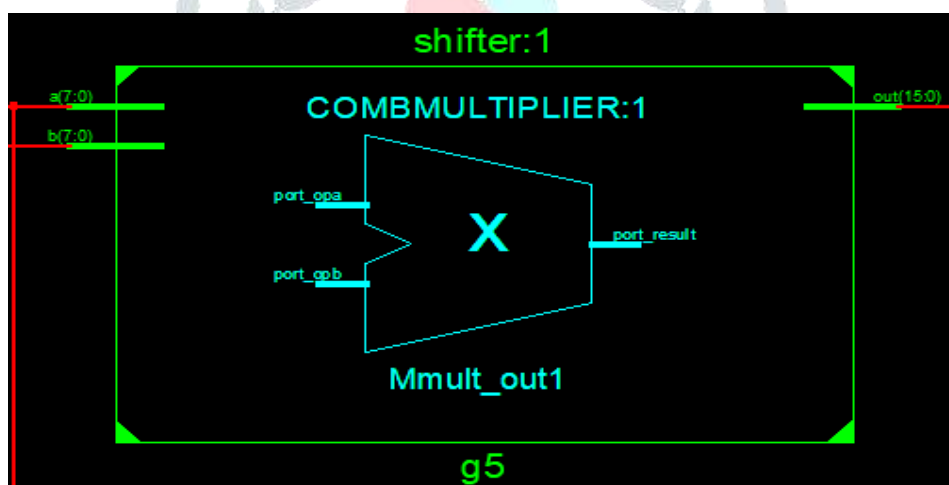


Figure 4: ROBA Shifter

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

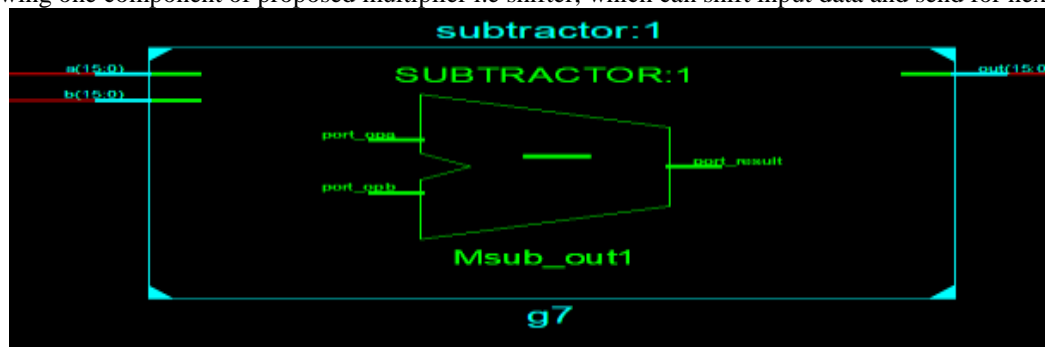


Figure 5: ROBA Subtractor

In figure 5, showing another component of proposed multiplier i.e Subtractor, which performs subtraction operation.

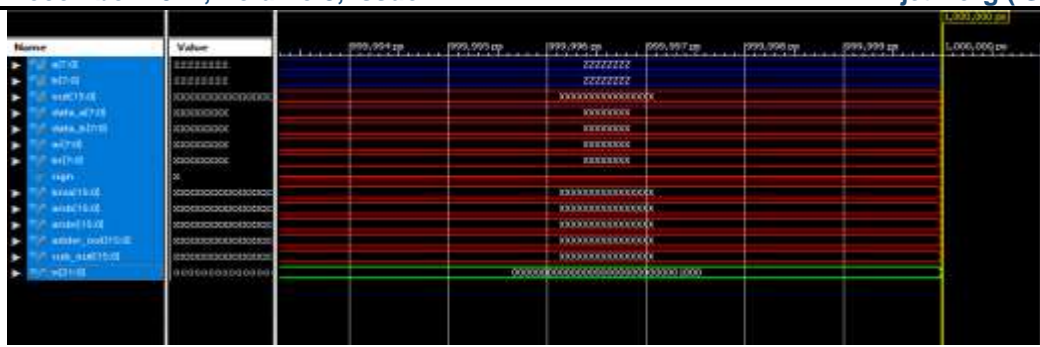


Figure 6: High impedance test bench bar

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

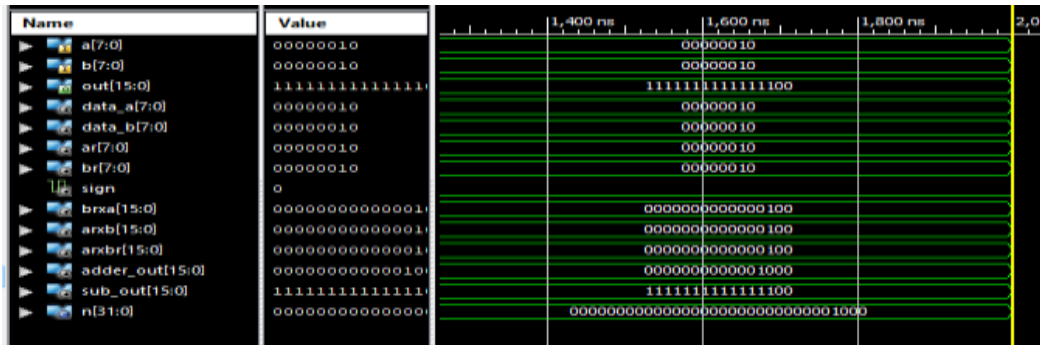


Figure 7: 8 Bit ROBA multiplier test bench in binary number

In figure 6 and 7, showing input a is 10 (2) and input b is 10(2) and output is 100(4)

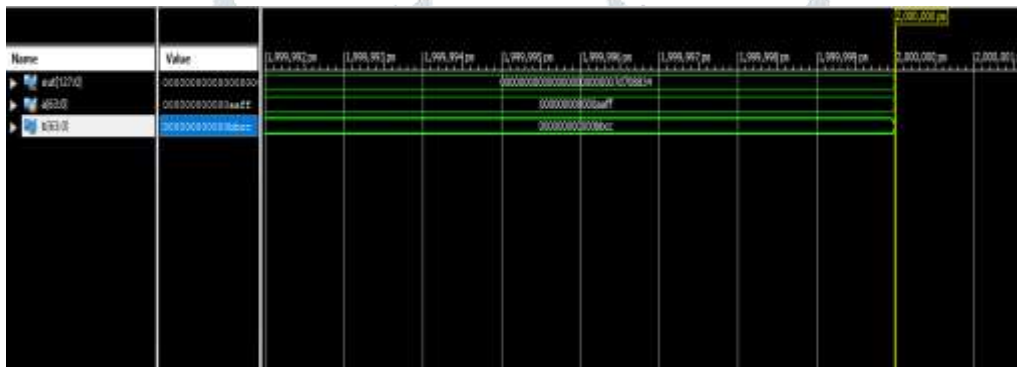


Figure 8: 64 Bit ROBA multiplier test bench in hexadecimal number

Figure 8 is showing the results validation, here input a is aaff and input b is bbcc and output is 7D708834

- Maximum combinational path delay: 19.104ns
- Timing constraint: Default path analysis
- Total number of paths / destination ports: 7802192 / 16
- Delay: 19.104ns (Levels of Logic = 30)
- Total: 9.104ns (11.734ns logic, 7.370ns route) (61.4% logic, 38.6% route)
- Total memory usage: 4396196 kilobytes

Table 1: Comparison with Previous and proposed work

Parameters	Previous work [1]	Proposed work
Type of Multiplier	32-bit ROBA	64-bit ROBA
Throughput (Speed)	22385 bits/s	436521 bits/s
Transmission time	4969.896 milli sec	9.10ns
Accuracy rate	70 %	90%
latency	29.983 ns	19.104ns

Table 1 is showing the simulation results of proposed work and previous work. It is clear from this table that proposed method can be calculate fast so that overall system speed will be improved. Therefore design and synthesis of ROBA multiplier using Xilinx verilog and find proposed multiplier is better than previous multiplier.

V. CONCLUSION

This paper design and analysis of 64-bit 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. Therefore implemented 64-bit ROBA multiplier gives significant improved performance. It achieved reduction of 10% area than existing 32-bit multiplier, save 60% power. Proposed multiplier is given 5% more accuracy than existing. Delay is significant constant so it can be say that it is improved than previous.

REFERENCES

- [1]. P. Lohray, S. Gali, S. Rangiseti 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, doi: 10.1109/CCWC.2019.8666472.
- [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. Rangiseti 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]. Ferozpur 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]. 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]. 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. Chakraborty, "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.