



## Efficient Design of FIR Filter using Modified ROBA Multiplier

<sup>1</sup>K. Geetha,<sup>2</sup>K. Avinash Kumar

<sup>1</sup>PG Scholar,<sup>2</sup>Associate Professor

<sup>1,2</sup>Department of Electronics and Communication Engineering,

<sup>1,2</sup>Avanthi Institute of Engineering and Technology, Tagarapavalasa, Vizianagaram, AP, India

**Abstract:** The increasing complexity of the DSP systems demanding higher computational performance in its architecture. But the traditional DSP arithmetic has limits in terms of speed of calculations. Moreover, in some applications, speed is more important than accuracy. To further enhance performance, approximate arithmetic circuits are designed with some loss of accuracy to reduce energy consumption and increase speed. These approximate circuits have been considered for error-tolerant applications. This paper proposes an FIR filter based on Modified Rounding Based Approximate Multiplier (ROBAM). In this Multiplier, the operands are rounded to the nearest exponent of two and used hybrid adder at the addition stage. This proposed Multiplier lead to simplification of multiplication operation, thus reducing area and increasing speed. As the Multiplier is the slowest element in the system, it affects the performance of the overall FIR filter. The proposed modified ROBA multiplier based FIR filter was compared with ROBA multiplier based FIR filters. The results have shown a significant reduction in the FIR filter area with proportional improvement in the multiplier speed. The Filter was synthesized and simulated using the Xilinx ISE environment.

**Index Terms - Approximation, ROBAM, FIR Filter, DSP**

### I. INTRODUCTION

In the field of the electronic industry, digital filters are used extensively. The noise ranges gradually increase by using analog filters for better noise performance can be obtained using digital filters compared to analog filters. At every intermediate step in digital filter transformation, able to perform noiseless mathematical operations. Our Design includes optimizing bit width and hardware resources without impacting the frequency response and output signal precision. Addition (or subtraction), Multiplication (generally of a signal by a constant), Time Delay, i.e. delaying a digital signal by one or more sample periods, are three basic mathematical operations used in digital filters. The coefficients are multiplied by fixed-point constants using additions, subtractions and shifts in a multiplier block. In VLSI Signal Processing, two types of digital filters are most widely used: FIR (finite impulse response) and IIR(infinite impulse response).

FIR indicates that the impulses are finite in this Filter, and phase, is kept linear in order to noise distortions and no feedback is used for such a filter. As compared to IIR, FIR is straightforward to design. Such types of FIR filters are used in DSP processors for high speed. In Digital Signal Processing, Multiplication and addition are of times required. A parallel prefix adder does a high-speed addition, and the better version of truncated Multiplier with fewer components reduces delay. In Digital Signal Processing, FIR filters define fewer bits designed by using finite precision.

One of the arithmetic operation multiplications is tiresome, so multipliers are the main components in arithmetic, signal and image processors. Signal processing and image processing consists of multiplying functions like multiply, accumulate, convolution and filtering. The operation rate of the multiplier unit impacts the execution time of a particular process. In Digital signal processing (DSP) algorithms, Multiplication takes more time than other operations, so a critical delay path is calculated for complete operation based on the delay required for the multiplication unit and measures algorithm performance. The most widely used operations in computer arithmetic are addition and Multiplication in approximate computing full-adder calls extensively analyzed for addition operation [1-3].

All DSP algorithms would need some form of Multiplication and Accumulation Operation. Most of the DSP algorithms need Multiplication and accumulation operation. MAC consists of the adder, Multiplier and accumulator. Usually, DSP adders are RCA and CSA. Generally, Multiplier multiplies the input values and passes the result to the adder. The adder adds to the previous result of the accumulator.

Contribution of Paper, Briefly introduction ROBAM applications in Section 1 and corresponding approximate multiplier Literature Survey is seeing in section 2. Architecture Explanation of Existing and Proposed ROBAM see in section 3 and section 4. Conventional FIR Filter and FIR Filter with Proposed ROBAM block diagram explanation see in section 5. Finally, the Results Explanation and conclusion see in sections 6 and section 7.

## II. LITERATURE SURVEY

M. J. Schulte et al. [4] presents hardware design FIR functions of reciprocal, square-root,  $2/\sup x/$ , and  $\log/\sub 2/(x)$  and produce exactly rounded results. In this paper used polynomial approximation in which the terms in the approximation are generated in parallel and then summed by using a multi-operand adder. To reduce the number of terms in the approximation, the input interval is partitioned into subintervals of equal size, and different coefficients are used for each subinterval. The coefficients used in the approximation are initially determined based on the Chebyshev series approximation for single-precision floating-point numbers, a design that produces exactly rounded results for all four functions has an estimated delay of 80 ns and a total chip area of 98 mm<sup>2</sup> in 1.0-micron CMOS technology. Allowing the results to have a maximum error of one unit in the last place reduces the computational delay by 5% to 30% and the area requirements by 33% to 77%.

R. Zendegani et al. [5] proposed high speed (HS) approximate ROBAM, and the approach is to round the operands to the nearest exponent of two. This way improves the speed and minimum energy consumption with small errors. The proposed ROBAM applies to both signed and unsigned multiplications. This paper proposed three architectures, One for unsigned Multiplication and the other two for signed multiplications. The Final proposed ROBAM improves the speed and performance compared to Existing approximate multipliers (EAM) and studied two image processing applications, i.e., image sharpening and smoothing. In this paper, we propose an estimated multiplier that is rapid yet vitality effective.

T. Su et al. [6] presents a formal approach to verify multipliers that approximate integer multiplication by output truncation. The method is based on extracting the polynomial signature of a truncated multiplier using algebraic rewriting. The proposed method consists of three basic steps: 1) determine the weights (binary encoding) of the output bits; 2) reconstruct the truncated Multiplier using functional merging and re-synthesis, and 3) construct the polynomial signature of the resulting circuit. The method has been tested on multipliers up to 256 bits with three truncation schemes: Deletion, D-truncation, and Truncation with Rounding. Experimental results are compared with the state-of-the-art SAT, SMT, and computer algebraic solvers.

D. De Caro et al. [7] presents piecewise polynomial interpolation as a well-established hardware function evaluation and minimizing polynomial coefficients word length to obtain either exact or faithful rounding at a reduced hardware cost. Using Integer Linear Programming (ILP), the proposed technique optimizes the polynomial coefficients taking into account all error components simultaneously. It has two benefits. Firstly, it can obtain exactly rounded approximations; secondly, for faithfully rounded interpolators, we avoid any overdesign due to pessimistic assumptions on error components, optimizing in this way the resulting hardware. The proposed ILP based algorithm requires a sufficient CPU time and is suited for approximations up to, maximum, 24 input bits. The results compare favourably with previously published data. We present synthesis results in 28 nm and 90 nm CMOS technologies.

E. Hosseini et al. [8] has proposed HP, and low power (LP) unsigned multiplication structure. The author first explains that the input bits of Multiplier are broken into several smaller groups of bits, and their Multiplication is calculated concurrently. The final product of Multiplication is generated after several rounds of the small group's results aggregation. A 32\*32-bit multiplier according to the proposed structure is designed in 180 nm CMOS standard library. The overall delay of proposed Multiplier is extremely low and is only 2.1ns. The power consumption is 41mW.

PentumSahasini et al. [9] propose a Modified rounding based approximate Multiplier (MROBAM), which is more accurate than the conventional ROBAM. The proposed ROBAM can be applied for both signed and unsigned numbers. Three hardware implementations are proposed in which one implementation for unsigned and two for signed operations. The accuracy of the proposed Multiplier is compared with the conventional rounding based approximate Multiplier (which are in  $2n$ ) where the modified rounding based approximate multiplier gives an exact output for the given inputs (irrespective of  $2n$ ) and various parameters like area, power delay, error significance, pass rates are been calculated and compared with conventional Multiplier where, MROBA gives better results and with the MROBA MAC unit is implemented. The design multiplier Proposed and Existing ROBAM in Xilinx 14.7.

A. Sai Sankalpa et al. [10] has proposed AM that's HS yet energy efficient. The approach is to round the operands to the closest exponent of two. This manner the procedure intensive a part of the Multiplication is omitted up speed and energy consumption at the worth of little error. The proposed AM is applicable to each signed and unsigned multiplications. The author proposed three hardware implementations of the AM that features one for the unsigned and 2 for the signed operations. The efficiency of the proposed Multiplier is evaluated by examining its performance with those of some approximate and exact multipliers using totally different style parameters. Additionally, the efficiency of the proposed approximate Multiplier is studied in two image process applications, i.e., image sharpening and smoothing.

S. Vahdat et al [11] has proposed truncation- and rounding-based scalable approximate Multiplier (TOSAM) is a scalable AM. The proposed Multiplier reduces the partial products (PP) by truncating each of the input operands based on their leading one-bit position. In this method used add, shift and small fixed-width multiplication operations resulting in large improvements in the energy consumption and area occupation compared to those of the EAM. To improve the total accuracy, input operands of the multiplication part are rounded to the nearest odd number. Results reveal that the proposed approximate Multiplier with a mean absolute relative error in the range of 11%–0.3% improves delay, area, and energy consumption up to 41%, 90%, and 98%, respectively, compared to those of the EAM. In this paper the proposed Multiplier applies to different applications like JPEG encoder, sharpening, and classification applications and also improves the accuracy of these applications compare to EAM.

P. Lohray et al [12] proposed AM can decrease the design complexity with an increase in performance and power efficiency for error avoid applications. In this paper, PP of the Multiplier is altered to introduce varying probability terms. The proposed AM is utilized in two variants of 16-bit multipliers. Synthesis results reveal that two proposed AM achieve power savings of 72% and 38%, respectively, compared to an EAM. They have better precision when compared to EAM. Mean relative error figures are as low as 7.6% and 0.02% for the proposed AM, which are better than the previous works. Performance of the proposed AM is evaluated with an image processing application, where one of the proposed models achieves the highest peak signal to noise ratio.

M.Pradeep Kumar et al. [13] proposed HS ROBAM by using HP Han-Carlson adder and it is multiplies only signed numbers. In this paper first explain the existing ROBAM and then designed Proposed ROBAM. The Design of Existing and proposed ROBAM by using Xilinx 14.7 and increase the speed of proposed ROBAM 4% compare to Existing ROBAM.

### III. EXISTING ROBA MULTIPLIER

The main idea behind the proposed approximate Multiplier is to make use of the ease of operation when the numbers are two to the power n (2n). To elaborate on the operation of the approximate Multiplier, first, let us denote the rounded numbers of the input of A and B by Ar and Br, respectively. The Multiplication of A by B may be rewritten as

$$A*B = ((A_r - A)*(B_r - B)) + (A_r*B) + (B_r*A) - (A_r*B_r) \tag{1}$$

The key observation is that the multiplications of Ar × Br, Ar × B, and Br × A may be implemented just by the shift operation. The hardware implementation of (Ar - A) × (Br - B), however, is rather complex. The weight of this term in the final result, which depends on differences of the exact numbers from their rounded ones, is typically small. Hence, we propose to omit this part from (2), helping simplify the multiplication operation. Hence, to perform the multiplication process, the following expression is used:

$$A*B = (A_r*B) + (B_r*A) - (A_r*B_r) \tag{2}$$

Thus, one can perform the multiplication operation using three shift and two addition/subtraction operations. In this approach, the nearest values for A and B in the form of 2n should be determined. When the value of A (or B) is equal to the 3 × 2(p-2) (where p is an arbitrary positive integer larger than one), it has two nearest values in the form of 2n with equal absolute differences that are 2p and 2p-1. While both values lead to the same effect on the accuracy of the proposed Multiplier, selecting the larger one (except for the case of p = 2) leads to a smaller hardware implementation for determining the nearest rounded value and hence, it is considered in this paper. It originates from the fact that the numbers in the form of 3 × 2(p-2) are considered as do not care in both rounding up and down simplifying the process, and smaller logic expressions may be achieved if they are used in the rounding up. It should be noted that contrary to the previous work where the approximate result is smaller than the exact result, the final result calculated by the ROBAM may be either larger or smaller than the exact result depending on the magnitudes of Ar and Br compared with those of A and B, respectively. Note that if one of the operands (say A) is smaller than its corresponding rounded value while the other operand (say B) is larger than its corresponding rounded value, then the approximate result will be larger than the exact result. This is due to the fact that, in this case, the multiplication result of (Ar - A) × (Br - B) will be negative. Since the difference between (2) and (3) is precisely this product, the approximate result becomes larger than the exact one. Finally, it should be noted the advantage of the proposed ROBAM exists only for positive inputs because in the two's complement representation, the rounded values of negative inputs are not in the form of 2n. Hence, we suggest that, before the multiplication operation starts, the absolute values of both inputs and the output sign of the multiplication result based on the inputs signs be determined and then the operation be performed for unsigned numbers and, at the last stage, the proper sign be applied to the unsigned result. Existing ROBAM architecture is shown in Fig. 1.

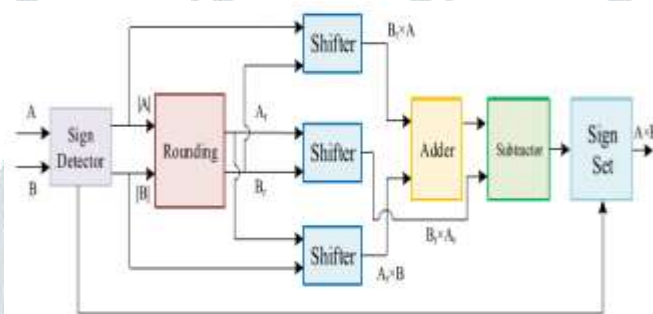


Fig. 1 Existing ROBAM architecture [5]

### IV. PROPOSED MODIFIED ROBA MULTIPLIER

In Existing ROBAM used ripple carry adder at addition stage but in proposed ROBAM used hybrid adder at adder stage. Modified or Hybrid adder is the faster addition technique generated by merging two adder designs that is carry select adder and kogge stone adder[14]. The carry-select adder is simple but rather fast. The carry select adder consist of two Ripple Carry Adder circuits and Multiplexer [15],[16],[17]. The main idea to save delay in carry select is to calculate all the bits and thus choosing only the correct one to obtain the result.

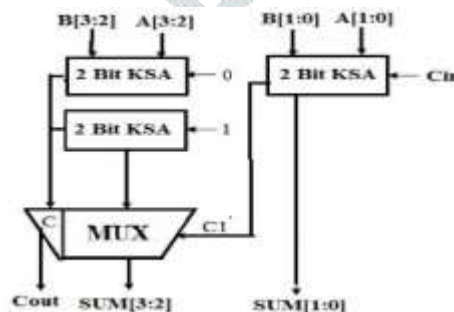


Fig. 2 Block Diagram of 4 bit Hybrid

Hence the two adders will calculate the sum and carry bits simultaneously and correct sum and carry bits are selected by multiplexer once the carry in bit is known. Since Ripple Carry Adder is used in the circuit the delay is much more increased so in place of RCA, Kogge Stone Adder is used which is a parallel prefix form of carry look ahead adder [18]. Kogge Stone generates carry bit fast. Hence in both adders the delay is reduced at the large extent so the properties of both the adder circuits can be used to design the high speed adder circuit. According to the previous block carry out signal a pair of 2:1 Multiplexers is used in order to select actual sum bit. The Architecture for 4 bit hybrid adder can be well understood from the diagram in figure given below Fig. 2.

From the diagram above, it is clear that first two bit operation is done using kogge stone addition method. And assuming two different carry signals i.e. either 0 or 1, all other paired block have been calculated twice by the same method [20]. Several multiplexers are needed in order to select actual sum as number of input bit increases. Proposed ROBAM with hybrid adder is shown in Fig. 3.



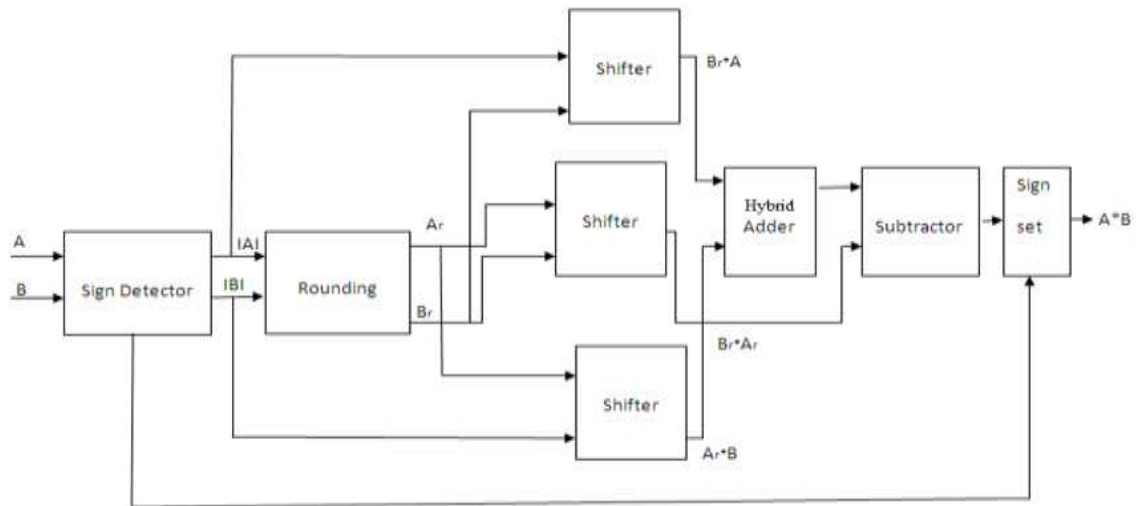


Fig. 3 Proposed Modified ROBA

**V. IMPLEMENTATION OF FIR FILTER**

**5.1 Conventional FIR Filter**

The conventional Design of the FIR filter is shown in Fig. 4. The implementation of an FIR requires three basic building blocks: Multiplication, Addition and Signal delay. FIR filter can be expressed as

$$y[n] = \sum_{k=0}^{N-1} b_k x[n - k]$$

Where N represents the filter order, y [n] is the output signal and b<sub>k</sub> represents the set of filter coefficients. If x[n] is the input signal applied, x [n - k] terms are referred as taps or tapped delay lines. In the conventional Design, present a simple structure of Multiplier for FIR filters. It performs Multiplication by generating partial products. If the multiplier digit is a 1, the multiplicand is simply copied down and represents the product. If the multiplier digit is a 0 the product is also 0. Therefore the area and delay will increase. It affects the performance of the FIR filter.

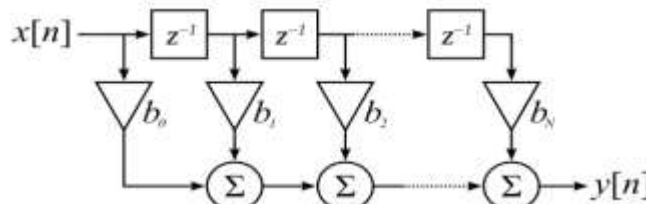


Fig. 4 Block diagram of FIR Filter

**5.2 FIR Filter Using Modified ROBA Multiplier**

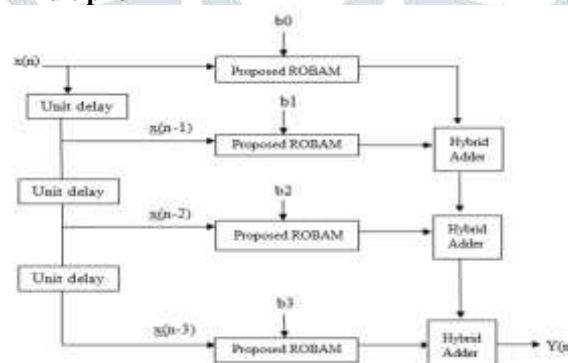


Fig.5 Block Diagram of FIR Filter using Proposed ROBA

As the Existing ROBA is the slowest element in the system, it will affect the performance of the FIR filter. So, Proposed ROBA is suggested since it reduces area and it is faster than other conventional ROBA. Block diagram of FIR Filter with Proposed ROBA is shown in Fig. 5.

**VI. RESULTS AND DISCUSSION**

Fig. 6 shows Simulation result for Proposed ROBA and Fig. 7 shows the implemented FIR filter using proposed ROBA. The simulations are performed by using Xilinx ISE simulator.

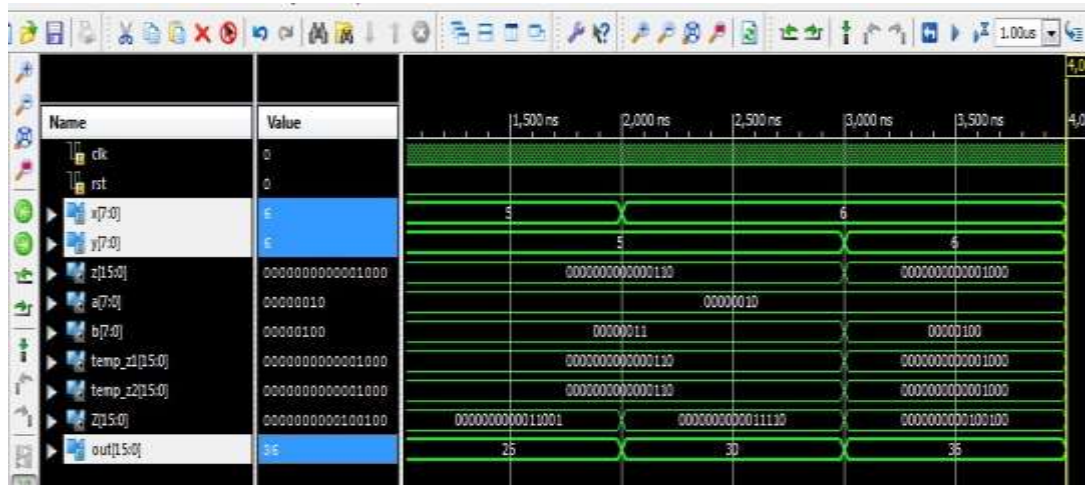


Fig. 6 Simulation result of Proposed ROBAM

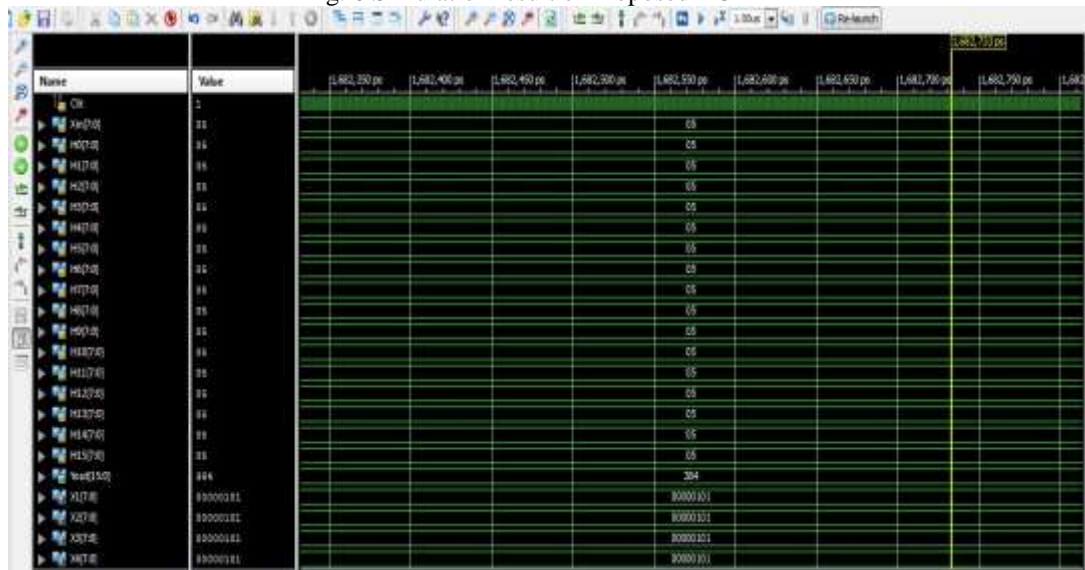


Fig. 7 Simulation result of FIR using Proposed ROBAM

When compared the results Proposed and Existing ROBAM with respect to area and delay following observations are made. From the table 1 the number of slices occupied by proposed ROBAM was very small when compared with the Existing ROBAM. This is due to the multiplication process has been significantly simplified by rounding the values to the nearest power of two. Even though this results a small error in the output value the area of the Multiplier has been drastically reduced. This reduced area results faster Multiplication which is more suitable for error tolerant DSP applications.

Table1: Comparison of Proposed ROBAM with Existing ROBAM

Name of the System	Proposed ROBAM	Existing ROBAM [5]
Number of Slices	100	192
Number of bounded IOB's	90	129
Delay (ns)	5.248	13.783

Next the FIR filter was implemented using Proposed ROBAM and Existing ROBAM. These FIR filters were compared with respect to area and delay. Clearly the table 2 shows that FIR filter with Proposed ROBA multiplier has occupied less area and a fractional reduction in the delay compared to FIR Filter using Existing ROBAM. The reduced area and delay are due to the fact that the Proposed ROBAM uses a good approximation technique which uses less number of components to perform the multiplication operation. Graphical representation of Table 1 and Table 2 is shown in Fig. 8 and Fig. 9.

Table 2: Comparison of FIR filters using both Proposed and Existing ROBAM

Name of the System	Proposed ROBAM	Existing ROBAM [5]
Number of Slices	2503	3010
Number of bounded IOB's	153	226
Delay (ns)	35.483	38.297

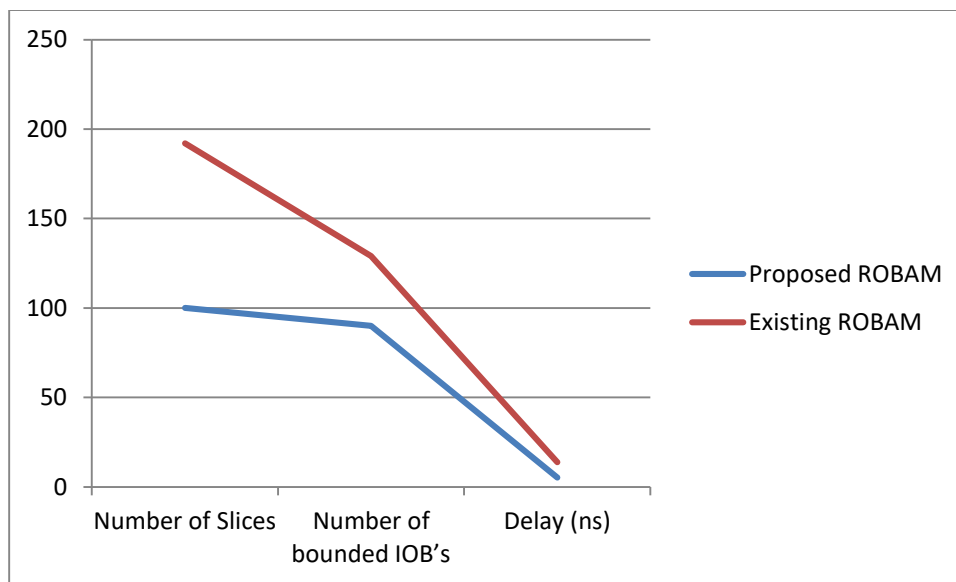


Fig. 8 Performance Comparison of Proposed and Existing ROBAM

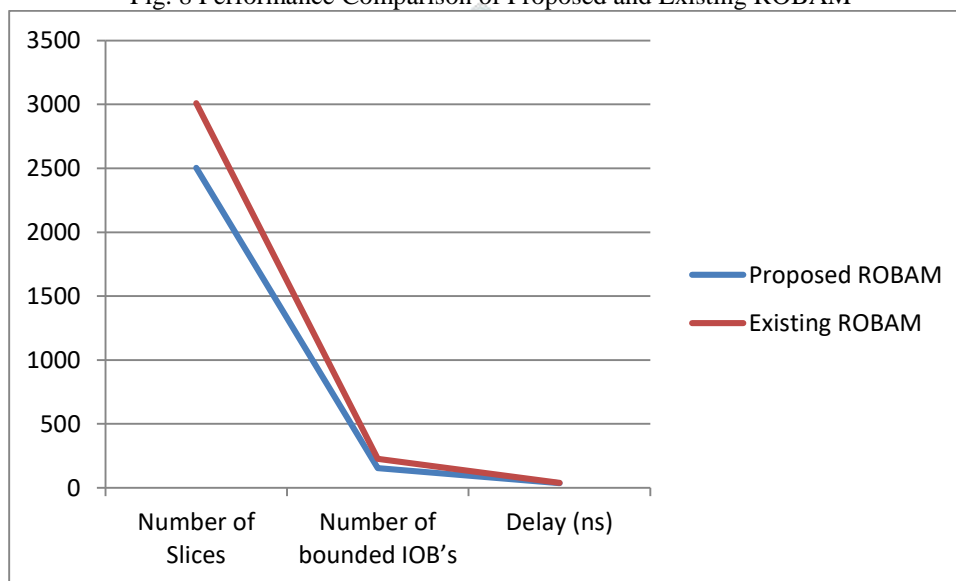


Fig. 9 Performance Comparison FIR Filter using Proposed and Existing ROBAM

## VII. CONCLUSION

Modified architecture of ROBAM based FIR filter implementation was proposed in the paper. The results show that the proposed ROBAM shows better performance in terms of area and delay. The Proposed ROBAM resulted in reduced area of the FIR filter when compare to the existing ROBAM. Moreover reduced area resulted in less delay. The system shows 62% speed reduced an occupying only 47.8 % the area of Existing ROBAM. The above results were obtained after simulating the FIR Filter using Xilinx ISE simulator at an operated volage of 1.0V.

## REFERENCES

- [1] V. Gupta, D. Mohapatra, S. P. Park, A. Raghunathan, and K. Roy, "IMPACT: IMPrecise adders for Low-Power Approximate Computing", Proc. of Int. Symp. On Low Power Electronics and Design (ISLPED). 1-3 Aug. 2011
- [2] S. Cheemalavagu, P. Korkmaz, K.V. Palem, B.E.S. Akgul, and L.N. Chakrapani, "A Probabilistic CMOS Switch and its Realization by Exploiting Noise," in Proc. IFIP-VLSI SoC, Perth, Australia, Oct 2005
- [3] H.R. Mahdiani, A. Ahmadi, S.M. Fakhraie, C. Lucas, "Bio-Inspired Imprecise Computational Blocks for Efficient VLSI Implementation of Soft-Computing Applications", IEEE Trans. on Circuits and Systems I: Regular Papers, Vol. 57, No. 4, pp. 850-862, April 2010
- [4] M. J. Schulte and E. E. Swartzlander, "Hardware designs for exactly rounded elementary functions," in IEEE Trans. on Computers, vol. 43, no. 8, pp. 964- 973, Aug. 1994
- [5] 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
- [6] 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
- [7] 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 Trans. on Circuits and Systems I: Regular Papers, vol. 64, no. 5, pp. 1187-1200, May 2017
- [8] E. Hosseini, M. Mousazadeh and A. Amini, "High-Speed 32\*32 bit Multiplier in 0.18um CMOS Process," 2018 25th Int. Conf. "Mixed Design of Integrated Circuits and System" (MIXDES), Gdynia, 2018, pp. 154-159

- [9] PentumSuhasini and Dr.J.Selvakumar," Modified Rounding Based Approximate Multiplier (MROBA) and MAC Unit Design for Digital Signal Processing", Int. Jour. of Pure and Applied Mathematics, vol. 118, No. 18, pp. 1539-1546, April 2018
- [10] A. Sai Sankalpa and C.Leela Mohan," Design and Synthesis Of Signed and Un-Signed Approximate Multiplier Using Rounding Based Approximation", International Journal of Research in Advent Technology, vol. 69, No. 6, June 2018
- [11] S. Vahdat, M. Kamal, A. Afzali-Kusha and M. Pedram, "TOSAM: An Energy-Efficient Truncation- and RoundingBased Scalable Approximate Multiplier," in IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 27, No. 5, pp. 1161-1173, May 2019
- [12] 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 Conf. (CCWC), Las Vegas, NV, USA, 2019, pp. 0420-0425.
- [13] M.Pradeep Kumar, A.S.Aparna, P.Pavan Kumar, D.SivaGangadhar Rao and N.M.Ramalingeswara Rao," Rounding Based Approximate (Roba) Multiplier", Journal of Emerging Technologies and Innovative Research (JETIR) (UGC Approval), vol.6, No. 4, pp. 448-454, April 2019
- [14] AritraMitra, Amit Bakshi, "Performance Improvement of a Modified Carry Select Adder", 24th IRF International Conference, Chennai, India, ISBN: 978-93-85465-07-9, pp 69-77, 3rd May 2015.
- [15] ShivaniParmar, Kirat Pal Singh, "Design of high speed hybrid carry select adder ", 3rd IEEE, International Advance Computing Conference (IACC), pp: 1656 – 1663, 2013.
- [16] PallaviSaxena, Urvashi Purohit, Priyanka Joshi," Analysis of Low Power, Area- Efficient and High Speed Fast Adder", International Journal of Advanced Research in Computer and Communication Engineering , ISSN : 2278-1021, Vol. 2, Issue 9, pp 3705- 3710, September 2013.
- [17] TheegalaRavinder Reddy, P.Anjiah , " Design of High Speed Hybrid Carry Select Adder", International Journal & Magazine of Engineering, Technology, Management and Research, ISSN No: 2348-4845, Volume No: 2 , Issue No: 7 , pp 1151-1156, July 2015.
- [18] AritraMitra ,Amit Bakshi, Bhavesh Sharma ,NileshDidwania "Design of a High Speed Adder",International Journal of scientific and Engineering Research, Volume 6, Issue 4, April 2015.
- [19] Jasmine Saini, Somya Agarwal, Aditi Kansal," performance, analysis and comparison of digital adders", IEEE, International Conference on Advances in Computer Engineering and Applications (ICACEA)IMS Engineering College, Ghaziabad, India, 2015.

