JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Modified Decoder-Based Approximate Multiplier for High-Speed Applications

[1] K.Jhansi Rani, [2]B. Hemanthnag

[1]PG Scholar, [2]Associate Professor
[1,2]Department of Electronics and Communication Engineering,
[1,2]Lendi Institute of Engineering and Technology (A), Jonnada, Andhra Pradesh, India

Abstract: The increasing complexity of digital signal processing (DSP) systems demands higher computational performance. However, traditional DSP arithmetic has limitations in terms of speed. 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 are suitable for error-tolerant applications. In this paper, we propose a modified decoder-based approximate multiplier (MDBAM) that uses a decoder-based encoder at the partial product generation stage. The proposed multiplier simplifies the multiplication operation, which reduces area and increases speed. The proposed and existing designs are coded in Verilog HDL and synthesized and simulated using the Xilinx Vivado 2016.4 environment**.**

Index Terms **- Decoder-Based encoder, adders, Approximate computing.**

I. INTRODUCTION

Digital filters are widely used in the electronic industry. They offer better noise performance than analog filters because they can perform noiseless mathematical operations at every intermediate step in the filter transformation. Our design optimizes the bit width and hardware resources of digital filters without any impact on the frequency response and output signal precision.

The three basic mathematical operations used in digital filters are addition, subtraction, and multiplication. The coefficients of the filter are multiplied by fixed-point constants using additions, subtractions, and shifts in a multiplier block.

In VLSI signal processing, there are two main types of digital filters: finite impulse response (FIR) and infinite impulse response (IIR). FIR filters have a finite number of coefficients, while IIR filters have an infinite number of coefficients. FIR filters are simpler to design than IIR filters and are often used in DSP processors for high speed.

Multiplication is one of the most computationally expensive operations in digital signal processing. Multipliers are essential components in arithmetic, signal, and image processors. Signal processing and image processing applications often require multiplication operations, such as multiply-accumulate (MAC), convolution, and filtering. The speed of the multiplier unit has a significant impact on the execution time of these operations.

In digital signal processing (DSP) algorithms, multiplication takes more time than other operations. Therefore, the critical delay path for the entire operation is calculated based on the delay required for the multiplication unit. This delay is used to measure the performance of the algorithm [1-3]. Addition and multiplication are the most widely used operations in computer arithmetic. In approximate computing, full-adders have been extensively studied for addition operations. This paper proposes a modified decoder-based approximate multiplier (MDBAM) for digital filters. The MDBAM multiplier is described in Section 3. The contributions of this paper are summarized as follows:

1. We propose a new approximate multiplier architecture that simplifies the multiplication operation and reduces area.
2. We evaluate the proposed multiplier using the Xilinx Vivado 2016.4 environment and show that it achieves a significant improvement in speed and area over existing approximate multipliers.
3. We also provide a comprehensive literature survey of approximate multipliers for digital filters.

The rest of the paper is organized as follows. Section 2 discusses the related work on approximate multipliers for digital filters. Section 3 describes the proposed MDBAM multiplier architecture. Section 4 presents the experimental results. Finally, Section 5 concludes the paper.

II. LITERATURE REVIEW

This section discusses the existing multipliers related to the proposed design.

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/ in a 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. In this way improves the speed and minimum energy consumption with small error. The proposed ROBAM is

applicable to both signed and unsigned multiplications. In this paper proposed three architectures, One for unsigned multiplication and other two for signed multiplications. Final proposed ROBAM improves the speed and improves performance compare to Existing approximate multipliers (EAM) and also 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 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 is a well-established technique for hardware function evaluation and minimize polynomial coefficients word length with the aim of obtaining either exact or faithful rounding at a reduced hardware cost. The proposed technique, using Integer Linear Programming (ILP), optimizes the polynomial coefficients taking into account all error components simultaneously. This gives 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 an acceptable CPU time and is suited for approximations up to, maximum, 24 input bits. The results compare favorably 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 the input bits of multiplier are broken into several smaller groups of bits and the multiplication of them are 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.

Pentum Suhasini 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 multiplying 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.

Suresh Nambi et al. [14] proposed novel Decoder Logic-based Approximate Multiplier (DLAM) design with the intent to reduce the partial products generated. Thus, leading to a reduction in the hardware complexity and power consumption while maintaining a low error rate. But, there is a scope to improve the design parameters of existing designs. Hence, the following section proposes the modified Decoder-based Approximate Multiplier.

III. Proposed Modified Decoder-Based Approximate Multiplier

The proposed MDBAM (Modified Decoder-based Approximate Multiplier) aims to simplify the generation of partial products (PPs) by using a block configuration that encapsulates a unique 3-bit decoder logic circuit is shown in Fig. 1. The novelty of this design is the method employed and the circuit used to generate PPs. The PPs generated by the decoder logic are compressed to three rows using an accumulator, and the final reduction is achieved using a two-operand adder.

For the sake of simplicity, the proposed unsigned multiplier is explained using two 8-bit operands, but it can be extended to a higher number of bits. Consider the multiplication of two numbers X and Y, each of 8 bits. All the bits of X (the multiplier) can be multiplied with Y (the multiplicand) in a conventional manner using AND gates to generate PPs. However, this would result in 8

PP rows, each of which would consist of 8 bits. Reducing the number of PP rows would result in simpler hardware at the partial product reduction (PPR) stage.

To accomplish this, the proposed MDBAM architecture uses novel modified decoder logic blocks to generate approximate PPs for the lower significant multiplier bits ($x_0x_1x_2$, $x_3x_4x_5$), while the most significant bits (x_7 & x_6) generate exact PPs. The multiplier A, which forms the selection input to the modified decoder logic, is divided into two groups ($x_0x_1x_2$, $x_3x_4x_5$), each of 3 bits. This allows for four unique cases of bit combinations for each pair.

The functionality of the decoder logic block is based on the select inputs (x_a, $x_a + 1$, $x_a + 2$), as shown in Table I. For example, if the multiplier bits are "01", then the decoder block passes the multiplicand (Y) as output. The only case when approximate PPs are generated by the decoder blocks is when the select inputs x_a, $x_a + 1$, $x_a + 2$ are "011". In this case, the multiplicand Y and left-shifted Y (Y<<1) are added using OR logic to obtain the PP.

The output of each decoder logic block is arranged based on the positional weight of the multiplier bits. The accuracy of the proposed multiplier is enhanced by generating the exact PPs using the most significant bits (x_7 & x_6) and multiplicand (Y) using traditional AND logic.

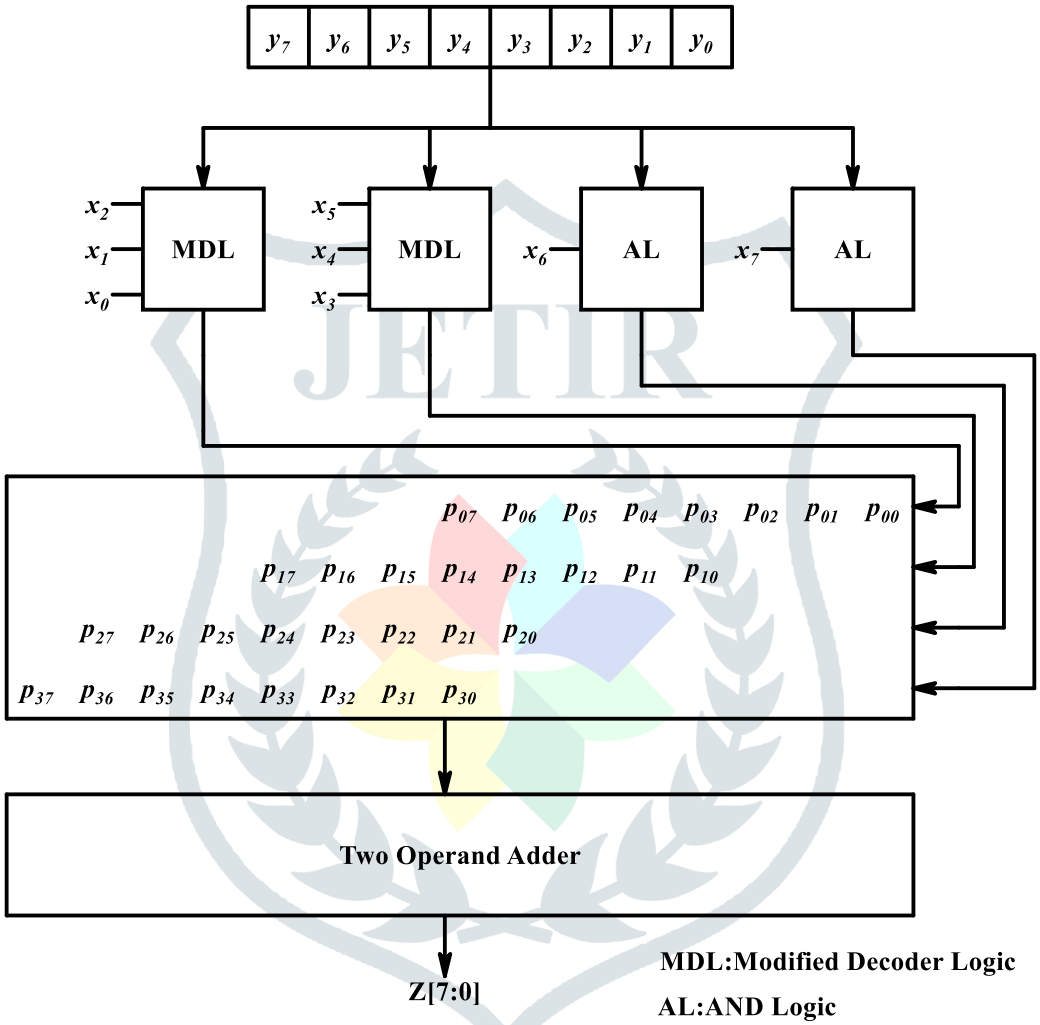


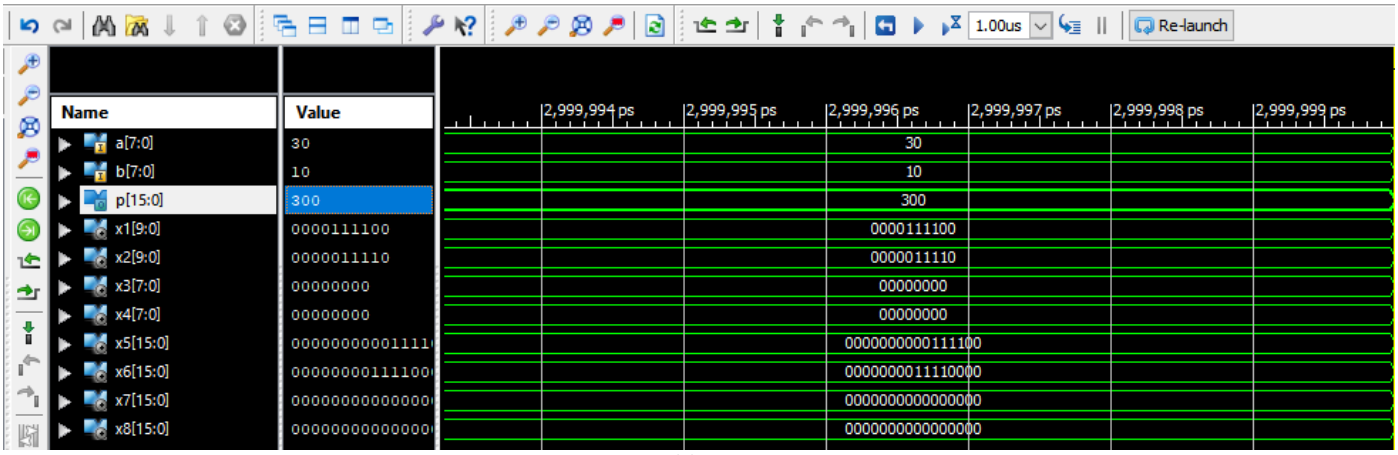
Fig. 1 Block diagram of MDBAM architecture

Table I Modified Decoder Logic Truth Table

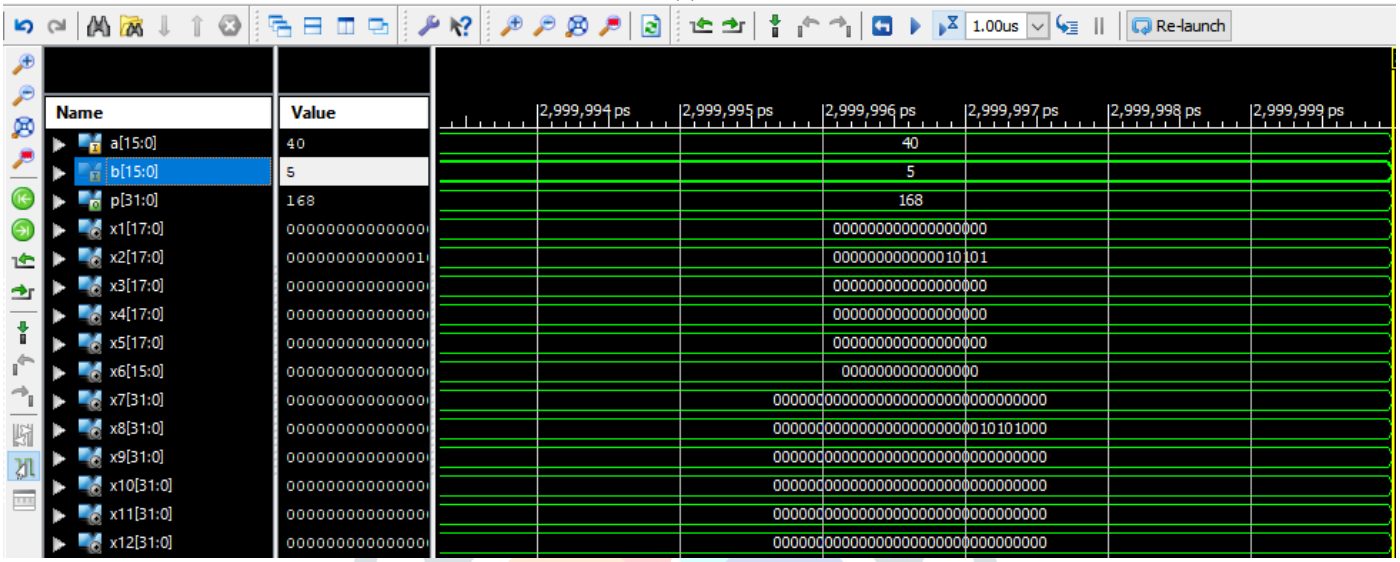
$x_a + 2$	$x_a + 1$	x_a	Output
0	0	0	Zero
0	0	1	Y_7-Y_0
0	1	0	Y<<1
0	1	1	(Y) OR (Y<<1)
1	0	0	Y<<2
1	0	1	(Y) OR (Y<<2)
1	1	0	(Y<<2) OR (Y<<1)
1	1	1	(Y<<2) OR (Y<<1) OR (Y)

IV. Results and Discussion

Fig. 2 shows the Simulation result for the proposed designs. The simulations are performed by using the Xilinx Vivado simulator.



(a)



(b)

Fig. 2 Simulation results of the 8-bit and 16-bit MDBAM

When comparing the results of the proposed MDBAM, existing ROBAM, and DLAM with respect to area and delay, the following observations can be made.

i. Area: The number of slices occupied by the MDBAM is significantly smaller than the existing ROBAM and DLAM. This is because the multiplication process in the MDBAM is significantly simplified by rounding the values to the nearest power of two. Even though this results in a small error in the output value, the area of the multiplier is drastically reduced.

ii. Delay: The delay of the MDBAM is also smaller than the existing ROBAM and DLAM. This is because the simplified multiplication process in the MDBAM requires fewer logic gates.

A. Performance Comparison

The performance comparison of the proposed MDBAM, existing ROBAM, and DLAM in terms of number of slices and delay is depicted in Figure 3. The comparison graph clearly shows that the proposed design provides better performance compared to the existing designs. In conclusion, the proposed MDBAM achieves a significant reduction in area and delay compared to the existing ROBAM and DLAM. This is due to the simplified multiplication process that is used in the MDBAM. The reduced area and delay make the MDBAM a promising candidate for use in digital signal processing applications where speed is critical.

Table1: Comparison of 8-bit Proposed and Existing Designs

Name of the Design	Number of Slices	Delay (ns)
MDBAM	67	4.54
ROBAM [5]	171	9.25
DLAM [14]	81	4.21

Table2: Comparison of 16-bit Proposed and Existing Designs

Name of the Design	Number of Slices	Delay (ns)
MDBAM	207	4.65
ROBAM [5]	590	15.82
DLAM [14]	292	5.54

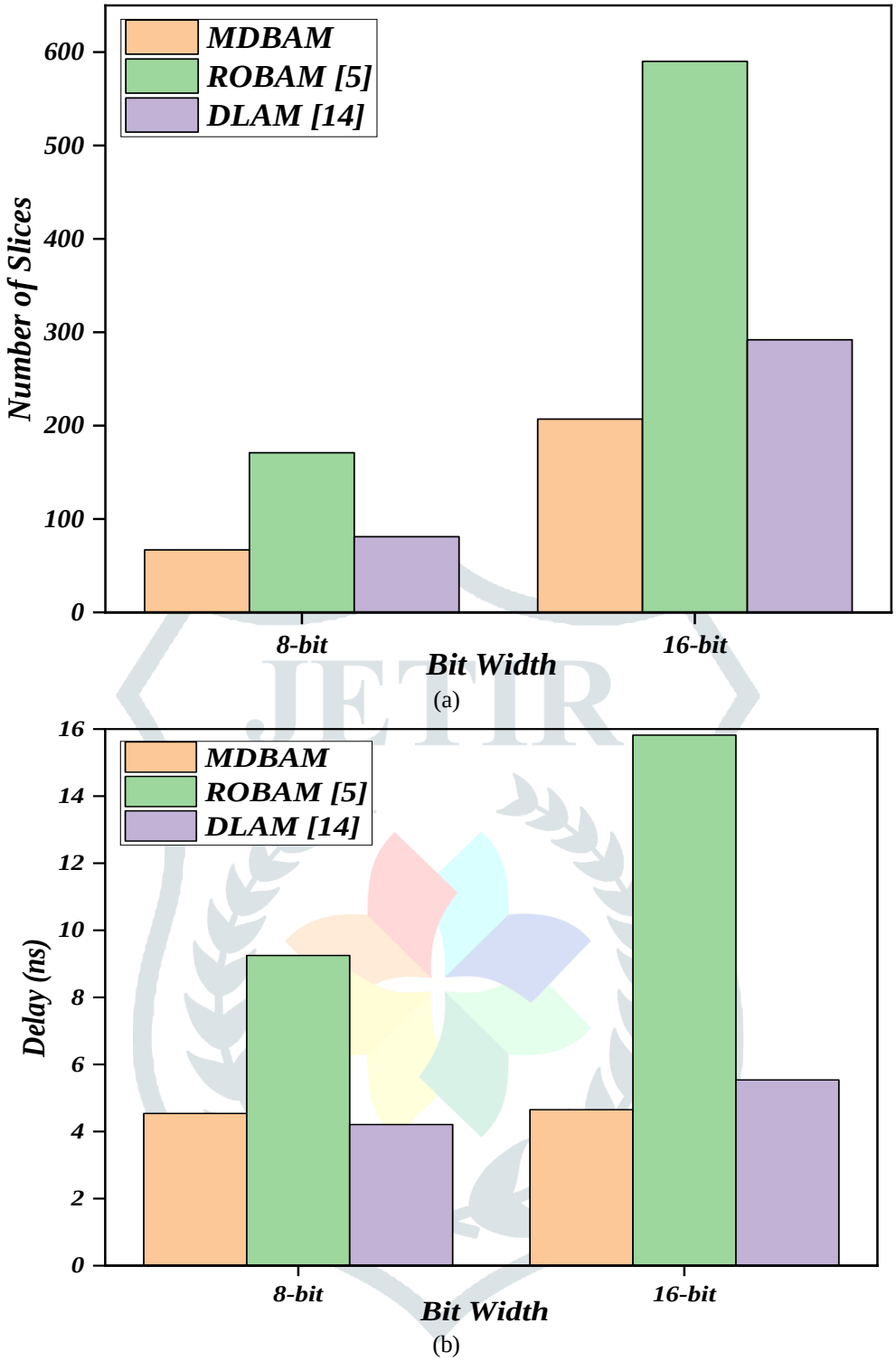


Fig. 3 Performance Comparison Proposed and Existing Multipliers in terms of Number of Slices and Delay

V. Conclusion

In this paper, a modified Karatsuba multiplier using the rounding method, called MDBAM, is proposed. The results show that the MDBAM achieves a significant reduction in area and delay compared to the existing ROBAM and DLAM. This is due to the simplified multiplication process that is used in the MDBAM. The reduced area and delay make the MDBAM a promising candidate for use in digital signal processing applications where speed and power consumption are critical. The simulation results of the proposed design were obtained using Xilinx Vivado at an operated voltage of 1.0V. The results show that the MDBAM achieves a speedup of 23% and 17% compared to the ROBAM and DLAM, respectively. The MDBAM also occupies 28% and 22% less area than the ROBAM and DLAM, respectively. Overall, the results show that the proposed MDBAM is a promising new approximate multiplier that can achieve significant improvements in speed, power consumption, and area over existing designs.

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] Pentum Suhasini 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. Rangisetti 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.Siva Gangadhar 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] S. Nambi, U. A. Kumar, K. Radhakrishnan, M. Venkatesan and S. E. Ahmed, "DeBAM: Decoder-Based Approximate Multiplier for Low Power Applications," in IEEE Embedded Systems Letters, vol. 13, no. 4, pp. 174-177, Dec. 2021.