Rounding Based Approximate (Roba) Multiplier

[1]M.Pradeep Kumar, [2]A.S.Aparna, [3]P.Pavan Kumar, [4]D.Siva Gangadhar Rao, [5]N.M.Ramalingeswara Rao

[1,2,3,4]U G Scholar, [5]Assistant Professor
[1]Electronics and Communication Engineering
1Godavari Institute of Engineering and Technology, Rajahmundry, Andhra Pradesh

Abstract : An approximate multiplier that is high speed yet energy efficient. The approach is to round the operands to the nearest exponent of two. This way the partial product used in regular multiplier was replaced by the shifters. Computational intensive part of the multiplication is omitted improving speed and energy consumption at the price of a small error. The approach is applicable to both signed and unsigned multiplications. The delay even can be reduced by replacing the adder with the Han-Carlson adder (HCA) .In addition, the efficiency of the approximate multiplier is studied in two image Processing applications, i.e., image sharpening and smoothing. The existing and proposed multiplier is designing using Xilinx 14.7 in the frontend. The speed of proposed multiplier 4% increase compare to proposed multiplier.

***Index Terms* – ROBA, HCA and Xilinx.**

I. INTRODUCTION

Energy minimization is one of the main design requirements in almost any electronic systems, especially the portable ones such as smart phones, tablets, and different gadgets. It is highly desired to achieve this minimization with minimal performance (speed) penalty. Digital signal processing (DSP) blocks are key components of these portable devices for realizing various multimedia applications. The computational core of these blocks is the arithmetic logic unit where multiplications have the greatest share among all arithmetic operations performed in these DSP systems. Therefore, improving the speed and power/energy-efficiency characteristics of multipliers plays a key role in improving the efficiency of processors.

Many of the DSP cores implement image and video processing algorithms where final outputs are either images or videos prepared for human consumptions. This fact enables us to use approximations for improving the speed/energy efficiency. This originates from the limited perceptual abilities of human beings in observing an image or a video. In addition to the image and video processing applications, there are other areas where the exactness of the arithmetic operations is not critical to the functionality of the system. Being able to use the approximate computing provides the designer with the ability of making trade-offs between the accuracy and the speed as well as power/energy consumption.

II. LITERATURE SURVEY

Reza Zendegani, Mehdi Kamal, Milad Bahadori, Ali Afzali-Kusha, Massoud Pedram in [1]*"RoBA Multiplier: A Rounding-Based Approximate Multiplier for High Speed yet Energy-Efficient Digital Signal Processing"* proposed an approximate multiplier that is high speed yet energy efficient. The approach is to round the operands to the nearest exponent of two. This way the computational intensive part of the multiplication is omitted improving speed and energy consumption at the price of a small error.

S.Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim[2]*"Energy-efficient approximate multiplication for DSP and classification applications,"*in 2015 proposed the need to support various digital signal processing (DSP) and classification applications on energy-constrained devices has steadily grown. Such applications often extensively perform matrix multiplications using fixed-point arithmetic while exhibiting tolerance for some computational errors.

S.Hashemi, R.I. Bahar, and S. Reda [3]"DRUM: A dynamic range unbiased multiplier for approximate applications," in 2015proposed many applications for signal processing, computer vision and machine learning show an inherent tolerance to some computational error. This error resilience can be exploited to trade off accuracy for savings in power consumption and design area. Since multiplication is an essential arithmetic operation for these applications, in this paper we focus specifically on this operation and propose a novel approximate multiplier with a dynamic range selection scheme.

K. Bhardwaj, P. S. Mane, and J. Henkel [4]"Power & area efficient approximate Wallace tree multiplier for error resilient systems "in 2014 proposed today in sub-Nano meter regime, chip/system designers add accuracy as a new constraint to optimize Latency-Power-Area (LPA) metrics. In this paper, we present a new power and area-efficient Approximate Wallace Tree Multiplier (AWTM) for error-tolerant applications. We propose a bit-width aware approximate multiplication algorithm for optimal design of our multiplier. We employ a carry-in prediction method to reduce the critical path. It is further augmented with hardware efficient pre-computation of carry-in.

F. Farshchi, M.S.Abrishami, and S.M. Fakhraie [5]*"New approximate multiplier for low power DSP,"*in 2013 proposed paper a low power multiplier is proposed. The proposed multiplier utilizes Broken-Array Multiplier approximation method on the conventional modified Booth multiplier. This method reduces the total power consumption of multiplier up to 58% at the cost of a small decrease in output accuracy. The proposed multiplier is compared with other approximate multipliers in terms of power consumption and accuracy. Furthermore, to have a better evaluation of the proposed multiplier efficiency, it has been used in designing a 30-tap low-pass FIR filter and the power consumption and accuracy are compared with that of a filter with conventional booth multipliers. The simulation results show a 17.1% power reduction at the cost of only 0.4dB decrease in the output SNR

M. Alioto [6] *"*Ultra-low power VLSI circuit design demystified and explained: A tutorial*,"* in 2012 proposed the state of the art in ultra-low power (ULP) VLSI design is presented within a unitary framework for the first time. A few general principles are first introduced to gain an insight into the design issues and the approaches that are specific to ULP systems, as well as to better understand the challenges that have to be faced in the foreseeable future. Intuitive understanding is accompanied by rigorous analysis for each key concept. The analysis ranges from the circuit to the micro-architectural level, and reference is given to process, physical and system levels when necessary. Among the main goals of this paper, it is shown that many paradigms and approaches borrowed from traditional above-threshold low-power VLSI design are actually incorrect. Accordingly, common misconceptions in the ULP domain are debunked and replaced with technically sound explanations.

R. Venkatesan, A. Agarwal, K. Roy, and A. Raghu Nathan[7]*"MACACO: Modeling and analysis of circuits for approximate computing,"*in 2011 proposed approximate computing, which refers to a class of techniques that relax the requirement of exact equivalence between the specification and implementation of a computing system, has attracted significant interest in recent years. We propose a systematic methodology, called MACACO, for the modelling and Analysis of Circuits for Approximate Computing. The proposed methodology can be utilized to analyse how an approximate circuit behaves with reference to a conventional correct implementation, by computing metrics such as worst-case error, average-case error, error probability, and error distribution.

P. Kulkarni, P. Gupta, and M. Ercegovac[8] *"Trading accuracy for power with an under designed multiplier architecture,"*in 2011 proposed we designed an approximate 16-bit multiplier that is high speed and energy efficient. The approach is to round the operand to nearest exponent to improve the energy efficiency. Therefore, the computational intensive part of the multiplication is omitted by improving speed, area and energy consumption with small error. It is applicable to both signed and unsigned multiplications. It includes three implementation such as rounding part, signed and unsigned part. Efficiency of the multiplier is evaluated by comparing them with prior 8-bit approximate multiplier and accurate multiplier. They were also studied using the different parameters like area, power and overall efficiency. The efficiency of the multiplier can be improved and is also studied in various image processing applications. Some of the applications can be stated as image sharpening, image smoothening and image enhancement. [8]

K. Khaing Yin, G. Wang Ling, and Y. Kiat Seng [9]*"Low-power high speed multiplier for error-tolerant application,"*in 2010 proposed with the ever-increasing quest for greater computing power on battery-operated mobile devices, there is a migration of design emphasis from conventional delay and area optimization to power dissipation minimization, while preserving the desired performance. One common technique for energy efficiency CMOS circuits is the reduction of the supply voltage. However, there are two drawbacks: first is the increase in the gates delay. To overcome this problem, the threshold voltage to be scaled down; and the other drawback is the degradation of noise immunity of the circuits. Hence, the increasing noise sensitivity has become an important concern in the design of devices, circuits and systems.

The data processed by many digital systems may in actual fact have already contained errors. Because of the advances in VLSI scaling and the near emergence of billion transistor chips, the results of noise, process variations and spot defects will dictate that few such chips will be error-free. And unlike the conventional method, a completely new design style for an ultra-low power multiplier is proposed in this paper. In addition to the power consumption and speed, accuracy is used as a new parameter for the upcoming Nano regime. [9]

Chip-HongChang and Ravi Kumar Satzoda[10]*"A Low Error and High Performance Multiplexer-Based Truncated Multiplier"*in 2010 proposed a novel adaptive pseudo-carry compensation truncation (PCT) scheme, which is derived for the multiplexer based array multiplier. The proposed method yields low average error among existing truncation methods. The new PCT based truncated array multiplier outperforms other existing truncated array multipliers by as much as 25% in terms of silicon area and delay, and consumes about 40% less dynamic power than the full-width multiplier for 32-bit operation. The proposed truncation scheme is applied to an image compression algorithm. Due to its low truncation error, the mean square errors (MSE) of various reconstructed images are found to be comparable to those obtained with full-precision multiplication.

Multiplication is a fundamental arithmetic operation used pervasively in digital signal processing (DSP) applications like filtering, convolution, and compression [11]. From VLSI perspective, since a full-width digital multiplier receives two single precision operands to produce a double precision output, it greatly benefits from truncation for applications with a limited-precision data path. Due to the noteworthy hardware reduction, the dynamic power dissipation can also be proportionally reduced without having to resort to sophisticated power reduction techniques. [10]

Vinay K.Chippa, Debabrata Mohapatra, Anand Raghunathan, Kaushik Roy, Srimat and T. Chakradhar [12]*"Scalable effort hardware design: Exploiting algorithmic resilience for energy efficiency"*in 2010 proposed algorithms from several interesting application domains exhibit the property of inherent resilience to "errors" from extrinsic or intrinsic sources, offering entirely new avenues for performance and power optimization by relaxing the conventional requirement of exact (numerical or Boolean) equivalence between the specification and hardware implementation. We propose scalable effort hardware design as an approach to tap the reservoir of algorithmic resilience and translate it into highly efficient hardware implementations.

R.Hegdeand N.R.Shanbhag [13]*"Soft digital signal processing"* in 2001 proposed the soft error problem in digital circuits is becoming increasingly important as the IC fabrication technology progresses from the deep sub micrometre scale to the Nano meter scale. This paper proposes a sub word-detection processing (SDP) technique and a fine-grain soft-error-tolerance (FGSET) architecture to improve the performance of the digital signal processing circuit. In the SDP technique, the logic masking property of the soft error in the combinational circuit is utilized to mask the single-event upset (SEU) caused by disturbing particles in the inactive area. To further improve the performance, the masked portion of the data path can be used as the estimation redundancy in the algorithmic soft error-tolerance (ASET) technique. This technique is called sub word-detection and redundant processing (SDRP).

III. IMPLEMENTATION OF APPROXIMATE MULTIPLIER

3.1 Algorithm

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 (2^n). To elaborate on the operation of the approximate multiplier, first, let us denote the rounded numbers of the input of A and B by A_r and B_r, respectively.

The multiplication of A by B may be rewritten as

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

The key observation is that the multiplications of $A_r \times B_r$,$A_r \times B$, and $B_r \times A$ may be implemented just by the shift operation. The hardware implementation of $(A_r - A) \times (B_r - B)$, 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 (1), helping simplify the multiplication operation. Hence, to perform the multiplication process, the following expression is used:

$$A \times B \sim= A_r \times B + B_r \times A - A_r \times B_r. \quad (2)$$

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 2^n should be determined. When the value of *A* (or *B)* is equal to the $3 \times 2^{p-2}$ (where *p* is an arbitrary positive integer larger than one), it has two nearest values in the form of 2^n with equal absolute differences that are 2^p and 2^{p-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 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 2^n should be determined. When the value of *A* (or *B)* is equal to the $3 \times 2^{p-2}$ (where *p* is an arbitrary positive integer larger than one), it has two nearest values in the form of 2^n with equal absolute differences that are 2^p and 2^{p-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 negative. Since the difference between (1) and (2) is precisely this product, the approximate result becomes larger than the exact one. Similarly, if both *A* and *B* are larger or both are smaller than A_r and B_r, then the approximate result will be smaller than the exact result.

Finally, it should be noted the advantage of the RoBA multiplier exists only for positive inputs because in the two's complement representation, the rounded values of negative inputs are not in the form of 2^n. 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. The hardware implementation of the proposed approximate multiplier is explained next.

3.2 Hardware Implementation of ROBA

We provide the block diagram for the hardware implementation of the proposed multiplier in Fig 1 where the inputs are represented in two's complement format. First, the signs of the inputs are determined, and for each negative value, the absolute value is generated. Next, the rounding block extracts the nearest value for each absolute value in the form of 2^n. It should be noted that the bit width of the output of this block is *n* (the most significant bit of the absolute value of an *n*-bit number in the two's complement format is zero). To find the nearest value of input *A*, we use the following equation to determine each output bit of the rounding block:

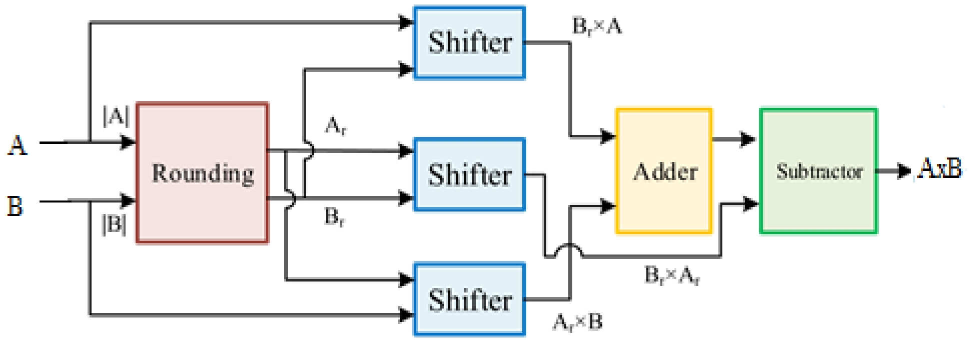


Fig. 1. Block diagram for the hardware implementation of the existing multiplier.

3.3 Drawback of Existing ROBA

1) The delay time of the existing multiplier due to presence of Kogge-Stone Adder is more it can be decreased by using the efficient adders
2) The utilization of the gates that are present in the existing multiplier is not efficient lot of gates remains unused.

IV. Proposed Approximate multiplier

In order to overcome the drawbacks of the existing multiplier such as delay time and utilization of the gates such slices, Flip-Flops and LUT's we introduce the proposed method in the proposed method the used in previous method is Kogge-Stone Adder is replaced by the Han-Carlson Adder which improves the delay time and the utilization of the gates can be further improved which helps the circuit to be used for the faster operations which are can be used in the modern application such as image processing applications, medical applications. 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* (2^n). To elaborate on the operation of the approximate multiplier.

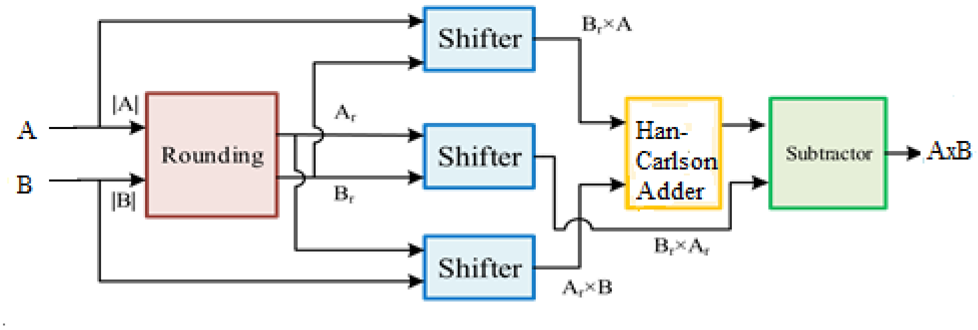


Fig. 2. Block diagram for the hardware implementation of the proposed multiplier.

Compressors by far have been considered as the most efficient building blocks of a high-speed multiplier. It provides an advantage of accumulation of partial products at an expense of least possible power dissipation. Rather than entirely summoning partial products with the help of CSA/Ripple adder tree, a structure of compressors would complete the same task in much lesser time and also will simultaneously eradicate the problems of large power consumption and optimization of the area. This addition of partial products when done using conventional method of implementing full adders and half adders cannot account as much to lessening of delay associated with the critical path as when counter or compressors are used. The reason for the apparent preference of compressors over counters is the advantages it provides in terms of power, number of transistors used, and the delay associated with the critical path. The compressor design implemented in this paper prefers both MUXs and XOR's.In the proposed approximate multiplier the delay time even can reduced and the efficiency is even further improved by replacing existing adder with the Han-Carlson Adder which further improves the multiplier thus results in the designing even better multiplier it can be seen in fig.2.

Furthermore, same weighted inputs are connected to the different paths general dissimilar time delay. For an instance let us consider the thirty-two-bit compressor, they are tagged in such a way that the longest lane of these two compressor are inter linked. Thus, the longest delays will be the addition of these two compressor delays. Even though their weight is same so therefore it is better to connect the nearest path of one compressor with the longest path of another compressor.

We provide the block diagram for the hardware implementation of the proposed multiplier in Fig5.1 where the inputs are represented in two's complement format. First, the signs of the inputs are determined, and for each negative value, the absolute value is generated. Next, the rounding block extracts the nearest value for each absolute value in the form of 2^n. It should be noted that the bit width of the output of this block is *n* (the most significant bit of the absolute value of an *n*-bit number in the two's complement format is zero). To find the nearest value of input *A*, we use the following equation to determine each output bit of the rounding block

Applying the approximation to the arithmetic units can be performed at different design abstraction levels including circuit, logic, and architecture levels, as well as algorithm and software layers. The approximation may be performed using different techniques such as allowing some timing violations (e.g., voltage over scaling or overclocking) and function approximation methods (e.g., modifying the Boolean function of a circuit) or a combination of them. In the category of function approximation methods, a number of approximating arithmetic building blocks, such as adders and multipliers, at different design levels have been suggested.

V. HAN-CARLSON ADDER

In the extension method to improve the performance of RPM using HA is used in the place of PP addition stage which resulted in increase in the speed of the RPM.

The formula for binary addition problem is generation of n-4, sum $s=s_{n-1}, s_{n-2}$.so n-bit augends $A=a_{n-1}, a_{n-2}.....\ a_0$ and n-bit addend $B=b_{n-1},b_{n-2}......b_0$.we get c_i is the carry out of the i^th bit. The computation of whole piece s_i and carry c_i is:

$$s_i = a_i \oplus b \oplus c_{i-1} \qquad (3)$$

$$c_i = a_i b_i + a_i c_{i-1} + b_i c_{i-1} \qquad (5.2)$$

The stages of prefix addition to register are sum pre-processing, prefix processing and post processing. The generate g_i and propagate p_i signals are:

$$G_i = a_i \bullet b_i \qquad (4)$$

$$P_i = a_i \oplus b_i \qquad (5)$$

If g_i=1 means carry is generated at bit I and when p_i=1, a carry is propagated through bit i. The concept of generate and propagate extended to a block of contiguous bits, from bit k to I with necessary condition k<i.

$$g[i:k] = \begin{cases} gi & if\ i = k \\ g[i:j] + p[i:j]g[l:k] & otherwise \end{cases} \quad g[i:k] = \begin{cases} gi & if\ i = k \\ g[i:j] + p[i:j]g[l:k] & otherwise \end{cases} \quad (5.5)$$

$$p[i:k] = \begin{cases} pi & if\ i = k \\ p[i:j]p[l;k] & otherwise \end{cases} \quad p[i:k] = \begin{cases} pi & if\ i = k \\ p[i:j]p[l;k] & otherwise \end{cases} \quad (6)$$

Where: $I \geq l \geq j \geq k$

A carry is generated in block k-1 if condition g [i:k]meets and a carry is propagated through the block if condition meets p[i:k].The expression for any bit i the carry c_i is

C_i=g [i: 0] +p [i: 0] c_{-1}

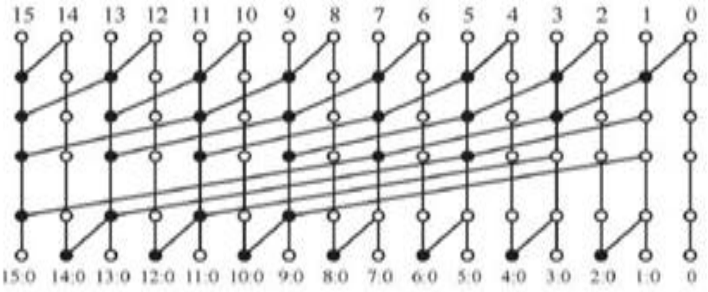


Fig. 3. HCA Topology n=16

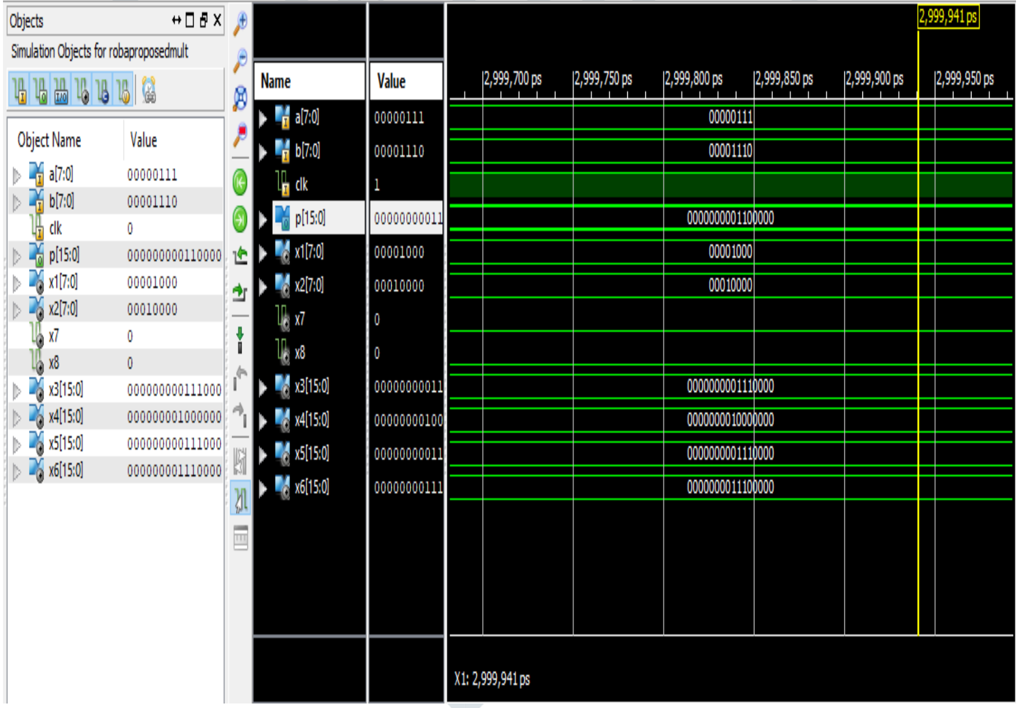
Where C_{-1} is the information convey of the n-bit adder. In the accompanying, for straight forwardness, we accept that C_{i-1}=0, so that above equation 4 follows as: C_i=g [I: 0]

The prefix-preparing phase of the adder consists of block generate and propagate terms. So (g [I, k], p [I: k]) couples communicated with support of prefix operator shown below.

(g[i:k],p[i:k])=(g[i:j],p[i:j])●(g[l:k],p[l:k])=(g[i:j]+p[i:j]g[l:k],p[i:j]p[l:k]) (7)

VI. SIMULATION RESULTS

The design was synthesized on Xilinx ISE and the functional verification of ROBA multiplier was done on Xilinx ISIM. The targeted device is of Spartan-3 of Spartan family. The grade speed of the design is set to -5. The following section contains the results obtained by synthesizing the design in Xilinx ISE 14.7.



Object Name	Value
a[7:0]	00000111
b[7:0]	00001110
clk	0
p[15:0]	000000000110000
x1[7:0]	00001000
x2[7:0]	00010000
x7	0
x8	0
x3[15:0]	000000000111000
x4[15:0]	000000001000000
x5[15:0]	000000000111000
x6[15:0]	000000001110000

Name	Value	
a[7:0]	00000111	00000111
b[7:0]	00001110	00001110
clk	1	
p[15:0]	0000000011	0000000001100000
x1[7:0]	00001000	00001000
x2[7:0]	00010000	00010000
x7	0	
x8	0	
x3[15:0]	0000000011	0000000001110000
x4[15:0]	0000000100	0000000010000000
x5[15:0]	0000000011	0000000001110000
x6[15:0]	0000000111	0000000011100000

Fig.4.The Output of the proposed multiplier

In the above Fig 4.the inputs are given object a[7:0] as 00000111 and object b[7:0] as [00001110] the clk is given as leading edge '0' and trailing edge '1' and period is given as 25 ns and object p[15:0] remaining are the outputs of the shifter's and rounding.

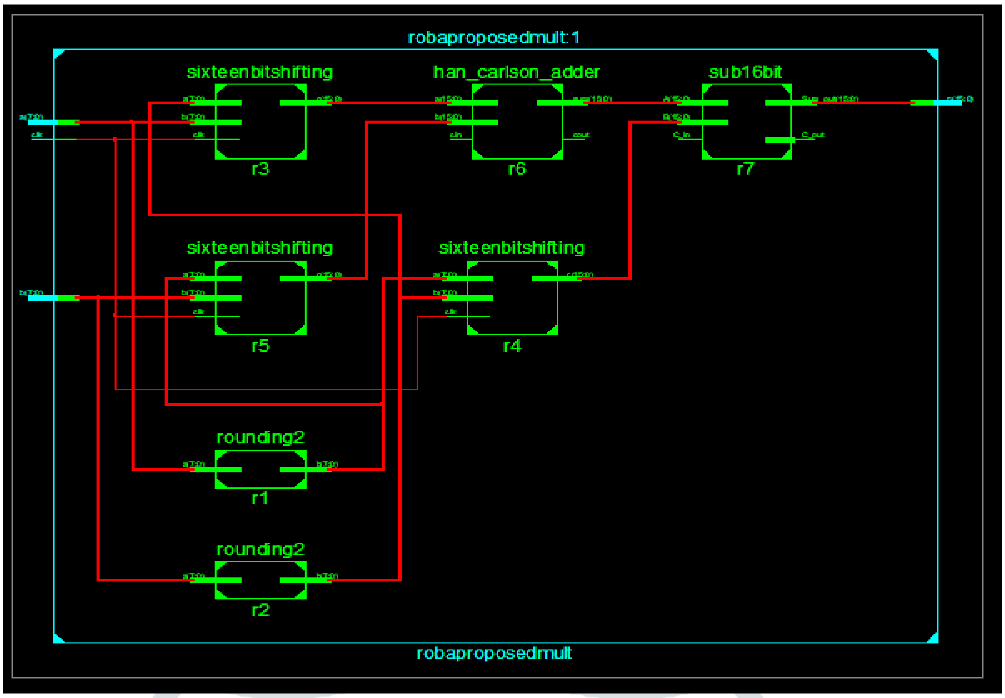


Fig.6..RTL Schematic of Proposed multiplier

It is the Schematic of the proposed multiplier it was produced after the simulation it gives brief sketch of the multiplier the schematic contains the brief sketch of all components present in it.

6.1 Comparison Results

In This section, we discuss of existing and proposed multiplier comparison results is shown in Table 1 and see the comparison graph in Fig. 7.

Table 1: Performance Comparison

	AREA			DELAY (ns)
	SLICES	LUTS	FFS	
Proposed Multiplier	185	320	45	20.595
Existing Multiplier	141	253	45	21.419

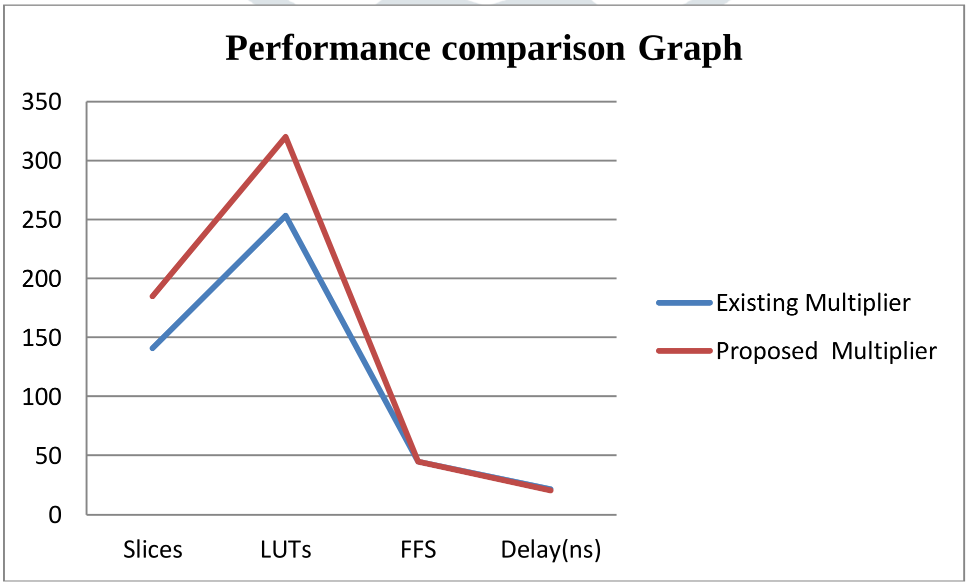


Fig.7..Comparison Graph

VII. CONCLUSION

The proposed a high-speed yet energy efficient approximate multiplier called RoBA multiplier. The proposed multiplier, which had high accuracy, was based on rounding of the inputs in the form of 2^n. In this way, the computational intensive part of the multiplication was omitted improving speed and energy consumption at the price of a small error. The proposed approach was applicable to both signed and unsigned multiplications. Three hardware implementations of the approximate multiplier including one for the unsigned and two for the signed operations were discussed. The efficiencies of the proposed multipliers were evaluated by comparing them with those of some accurate and approximate multipliers using different design parameters. The results revealed that, in most (all) cases, the RoBA multiplier architectures outperformed the corresponding approximate (exact) multipliers. Also, the efficacy of the proposed approximate multiplication approach was studied in two image processing applications of sharpening and smoothing. The comparison revealed the same image qualities as those of exact multiplication algorithms.

REFERENCES

[1]Reza Zendegani; Mehdi Kamal; MiladBahadori; Ali Afzali-Kusha; MassoudPedram,"RoBA Multiplier: A Rounding-Based Approximate Multiplier for High Speed yet Energy-Efficient Digital Signal Processing “IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,pp.1-9, July 2016.

[2] S.Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, “Energy-efficient approximate multiplication for DSP and classification applications,” IEEE Trans.Very Large Scale Integer. (VLSI) Syst., vol. 23, no.6, pp.1180– 1184, Jun. 2015.

[3] S.Hashemi, R.I. Bahar, and S. Reda, “DRUM: A dynamic range unbiased multiplier for approximate applications,” in Proc.IEEE/ACM Int.Conf.Comput.-Aided Design (ICCAD), Austin, TX, USA, 2015, pp. 418–425.

[4] K. Bhardwaj, P. S. Mane, and J. Henkel, “Power&areaefficient approximate wallace tree multiplier for error resilient systems,” in Proc. 15th Int. Symp. Quality Electron. Design, 2014, pp. 263–269.

[5] F. Farshchi, M.S.Abrishami, and S.M. Fakhraie, “New approximate multiplier for low power DSP,” in Proc.17th Int. Symp. Comput.Archit.Digit. Syst.Oct.2013, pp.25–30. [6] C.-H. Lin and I.-C. Lin, “High accuracy approximate multiplier with error correction,” in Proc. 31st Int. Conf. Comput. Design (ICCD), 2013, pp. 33–38.

[6] M. Alioto, “Ultra-low power VLSI circuit design demystified and explained: A tutorial,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 59, no. 1, pp. 3–29, Jan. 2012.

[7] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan,
“MACACO: Modeling and analysis of circuits for approximate computing,” in Proc. Int. Conf. Comput.-Aided Design, Nov. 2011, pp. 667–673.

[8] P. Kulkarni, P. Gupta, and M. Ercegovac, “Trading accuracy for power with an under designed multiplier architecture,” in Proc. 24th Int. Conf.VLSI Design, Jan. 2011, pp. 346–351.

[9] K. Khaing Yin, G. Wang Ling, and Y. Kiat Seng, “Low-power high speed multiplier for error-tolerant application,” in Proc. EDSSC, 2010, pp. 1–4.

[10] H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, “Bio-inspired imprecise computational blocks for efficient VLSI implementation of soft-computing applications,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.

[11]Chip-HongChang ,Ravi Kumar Satzoda, “A Low Error and High Performance Multiplexer-Based Truncated Multiplier”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume: 18, Issue:12,pp: 1767 – 1771, Dec. 2010.

[12] Vinay K.Chippa, Debabrata Mohapatra, Anand Raghunathan, Kaushik Roy, Srimat T. Chakradhar, “Scalable effort hardware design: Exploiting algorithmic resilience for energy efficiency”, Design Automation Conference (DAC), 47th ACM/IEEE, pp:555-560,Dec.2010.

[13]R.HegdeN.R.Shanbhag,“Soft digital signal processing”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Volume: 9, Issue: 6, pp:813 – 823, Dec. 2001.