

DESIGN AND SYNTHESIS OF SIGNED AND UNSIGNED APPROXIMATE MULTIPLIER USING ROUNDING BASED APPROXIMATION

¹Dr.Dola Sanjay.S, ²Dr.B.S.Satish, ³M.Tejaswi

^{1,2} Professor in Dept. Of ECE in Ramachandra College of Engineering, Vatluru, Andhra Pradesh 534001.

³ M.Tech (VLSI) in Dept. Of ECE in Ramachandra College of Engineering, Vatluru, Andhra Pradesh 534001.

Abstract: In this paper, we tend to propose an approximate multiplier that's high speed yet energy efficient. The approach is to round the operands to the closest exponent of 2. 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 approach is applicable to each signed and unsigned multiplications. We tend to propose three hardware implementations of the approximate multiplier 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.

Index Terms - Accuracy, approximate computing, energy efficient, error analysis, high speed, multiplier.

INTRODUCTION

Energy minimization is one among the most style necessities in any electronic systems, particularly the moveable ones like mobile phones, tablets, and totally different gadgets [1]. It's extremely desired to realize this minimization with the least performance (speed) penalty [1]. Digital signal processing (DSP) blocks are key elements of those moveable devices for realizing varied multimedia system applications. The procedure core of those blocks is that the arithmetic logic unit wherever multiplications have the best share among all arithmetic operations performed in these DSP systems [2]. Therefore, rising the speed and power/energy-efficiency characteristics of multipliers plays a key role in rising the efficiency of processors. Many of the DSP cores implement image and video process algorithms wherever final outputs are either pictures or videos made for human consumptions. This truth permits us to use approximations for raising speed/energy efficiency. This originates from the restricted sensory activity skills of individuals to perceive a picture or a video. Additionally to the image and video process applications, there are different areas wherever the truth of the arithmetic operations isn't crucial to the performance of the system (see [3], [4]). The power to use the approximate computing provides the designer with the ability to create tradeoffs between the accuracy and also the speed additionally as power/energy consumption [2], [5]. Applying the approximation to the

arithmetic units are often performed at completely different style abstraction levels together with circuit, logic, and design

levels, additionally as algorithmic rule and software system layers [2]. The approximation could also be performed by taking completely different techniques like permitting some temporal order violations (e.g., the voltage over scaling or over clocking) and performance approximation strategies (e.g., modifying the Boolean perform of a circuit) or both of them [4], [5]. within the class of performing approximation strategies, a variety of approximating arithmetic building blocks, like adders and multipliers, at completely different style levels are instructed (see [6]–[8]). During this paper, we tend to concentrate on proposing a high-speed low power/energy however approximate multiplier applicable for error resilient DSP applications. The proposed approximate multiplier, that is additionally space efficient, is built by modifying the standard multiplication approach at the algorithmic program level assumed as rounded input values. We tend to call as this rounding- based approximate (Roba) multiplier. The proposed multiplication approach is applicable to each signed and unsigned multiplications that 3 optimized architectures are given. The efficiencies of those structures are assessed by examining the delays, power and energy consumptions, energy-delay product EDP), and areas with those of some approximate and correct (exact) multipliers. The contributions of this paper will be summarized as follows:

- 1) Presenting a novel theme for Roba multiplication by modifying the standard multiplication approach
- 2) Describing three hardware architectures of the proposed approximate multiplication theme for a sign and unsigned operations. The remainder of this paper is organized as follows.

LITERATURE SURVEY

In this section, a number of the previous works within the field of the approximate multiplier are briefly reviewed. In [3], an approximate multiplier and an approximate adder found a method named broken-array multiplier (BAM) was the multiplier. By applying the BAM approximation methodology of [3] to the standard modified Booth multiplier factor, an approximate signed Booth multiplier was given in [5]. The approximate multiplier provided power consumption savings kind

28th to 58.6% and space reductions from 19.7% to 41.8% for various word lengths as compared with multiplier Booth 31. Kulkarni et al. [6] advised an approximate multiplier consisting of a variety of 2×2 inaccurate building blocks that saved 31.8%– 45.4% over an exact multiplier. An approximate signed 32-bit multiplier for speculation functions in pipelined processors was designed in [7]. it had been two hundred times quicker than a full-adder-based tree multiplier whereas having a chance of error of around 14%. In [8], an error-tolerant multiplier, that computed the approximate result by dividing the multiplication into one exact and one approximate part, was introduced, in which the accuracies for various bit widths were

reportable. In the case of a 12-bit multiplier, a power saving of over 50% was reported. In [9], two approximate 4:2 compressors for utilizing in a very regular Dadda multiplier were designed and analyzed. The utilization of approximate multipliers in image process applications that results in reductions in power consumption, delay, and semiconductor unit count compared with those of a particular multiplier style has been mentioned within the literature. In [10], an accuracyConfigurable multiplier design (ACMA) was advised for error-resilient systems to extend its output, the ACMA created use of a method referred to as carry-in prediction that worked supported a precipitation logic. Compared with the precise one, the projected approximate multiplication resulted in nearly 50% reduction within the latency by reducing the important path. Also, Bhardwaj et al. [11] had given an approximate Wallace tree multiplier (AWTM). Again, it invoked the carry-in prediction for the reduction of the vital path. In this work, AWTM was utilized in a time period benchmark image application showing concerning 40% and 30% reductions within the power and space, severally, without any image quality loss compared with the case of using a multiplier Wallace tree multiplier factor (WTM) structure. In [12], approximate unsigned multiplication and division supported an approximate log of the operands are proposed. Within the proposed multiplication, the summation of the approximate logarithms determines the results of the operation. Hence, the multiplication is simplified to some shift and adds operations. In [13], a technique for increasing the accuracy of the multiplication approach of [12] was it had been. it absolutely was it was the decomposition of the input operands. This methodology significantly improved the common error at the worth of increasing the hardware of the approximate multiplier factor by based on twice. In [16], a dynamic section methodology (DSM) is given, that performs the multiplication operation on an m-bit segment starting from the leading one bit of the input operands. A dynamic vary unbiased multiplier factor (DRUM) multiplier, that selects an m-bit bit of ranging from the leading one little bit of } the input operands and sets the smallest amount significant bit of the truncated values to 1, has been proposed in [17]. In this structure, the truncated values are increased and shifted to the left to come up with the final output. In [18], an approximate 4×4 WTM has been projected that uses an inaccurate 4:2 counter. Additionally, the miscalculation correction unit for correcting the outputs has been advised. To construct larger multipliers, this 4×4 inaccurate Wallace multiplier may be utilized in an array structure. Most of the previous proposed approximate multipliers are depends on either modifying the structure or complexness reduction of a specific accurate multiplier. In this paper, similar to [12], we have a tendency to propose activity the approximate multiplication through simplifying the operation. The distinction between our work and [12] is that, though the principles in eachwork are nearly similar for unsigned numbers, the mean error of our proposed approach is smaller. Additionally, we propose some approximation techniques once the multiplication is performed for signed numbers.

PROPOSED APPROXIMATE MULTIPLIER

Multiplication rule of Roba multiplier

The main plan behind the proposed approximate multiplier is to create use of the benefit of operation whenever the numbers are 2 to the power n (2n). To elaborate on the operation of the approximate multiplier first, allow us to denote the rounded numbers of the input of A and B by Ar and Br, individually. The multiplication of A by B is also rewritten as

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

The key observation is that the multiplications of Ar × Br, Ar × B, and Br × A could also be enforced simply by the shift operation. The hardware implementation of (ArA) 8 × (BrB), however, is very complicated. the weight of this term within the final result, that depends on variations of the precise numbers from their rounded ones, is usually little. Hence, we tend to propose to omit this part from (1), serving to modify the multiplication operation. Hence, to perform the multiplication method, the subsequent expression is used:

$$A \times B \sim = A_r \times B + B_r \times A - A_r \times B_r \tag{2}$$

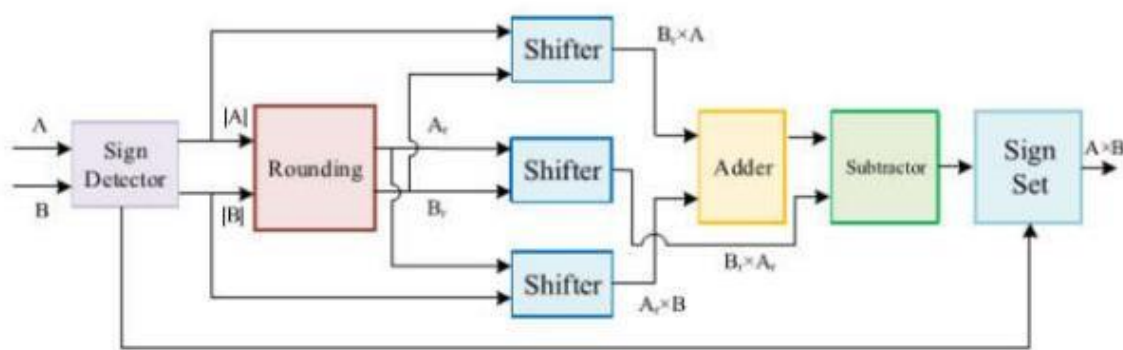


Fig.1:Block diagram for the hardware implementation of the proposed multiplier

Thus, one will perform the multiplication operation using three shift and two addition/subtraction operations. during this approach, the closest values for A and B within the kind of 2n ought to be determined. When the worth of A (or B) is adequate the $3 \times 2^{p-2}$ (where p is an absolute positive number larger than one), it's 2 nearest values within the kind of 2n with equal absolute variations that are 2^p and 2^{p-1} . Whereas each value result in a constant impact on the accuracy of the proposed multiplier, choosing the larger one (except for the case of p = 2) ends up in a smaller hardware implementation for determining the closest rounded worth, and hence, it is considered in this paper. It originates from the actual fact that the numbers within the sort of $3 \times 2^{p-2}$ are thought of as don't care in each rounding error up and down simplifying the

method, and smaller logic expressions are also achieved if they're utilized in the rounding error up. The main exception is for 3, that during this case, two is taken into account as its nearest price within the proposed approximate multiplier. It ought to be noted that contrary to the previous work wherever the approximate result's smaller than the precise result, the final result calculated by the Roba multiplier is also either larger or smaller than the precise result counting on the magnitudes of A_r and B_r compared with those of A and B , severally. Note that if one amongst the operands (say A)

is smaller than its corresponding rounded price whereas the opposite operand (say B) is larger than its corresponding rounded value, then the approximate result is going to be larger than the precise result. This can be due to the actual fact that, in this case, the multiplication result of $(A_r - A) \times (B_r - B)$ are going to be negative. Since the distinction between (1) and (2) is exactly this product, the approximate result becomes larger than the precise one. Similarly, if each A and B is larger or each is smaller than A_r and B_r , then the approximate result is going to be smaller than the precise result. Finally, it ought to be noted the advantage of the proposed Roba multiplier exists just for positive inputs as a result of within the two's complement illustration, the rounded values of negative inputs aren't within the kind of 2^n . Hence, we advise that, before the multiplication operation starts, absolutely the values of each input and therefore the output sign of the multiplication result depends on the inputs signs be determined and so the operation be performed for unsigned numbers and, at the last stage, the correct sign be applied to the unsigned result. The hardware implementation of the proposed approximate multiplier is explained next.

Hardware Implementation of Rounding based approximate multiplier

Primarily based on (2), we provide the block diagram for the hardware implementation of the proposed multiplier in Fig. 1 wherever the inputs are depicted in two's complement format. First, the signs of the inputs are determined, and for every negative value, the absolute value is generated. Next, the rounding block extracts the closest price for every exact value within the type of 2^n . It ought to be noted that the bit dimension of the output of this block is n (the most significant little bit of a price of an n -bit range within the two's complement format is zero). To find the closest price of input A , we have a tendency to use the subsequent equation to see every output little bit of the rounding error block:

Input 1 ($A_r \times B + B_r \times A$)	Input 2 ($A_r \times B_r$)	Output
000...11...xxx	000...10...000	000...01...xxx
000...11...xxx	000...01...000	000...10...xxx
000...10...xxx	000...01...000	000...01...xxx

$$\begin{aligned}
 A_r[n-1] &= \overline{A[n-1]} \cdot A[n-2] \cdot A[n-3] \\
 &\quad + A[n-1] \cdot \overline{A[n-2]} \\
 A_r[n-2] &= (\overline{A[n-2]} \cdot A[n-3] \cdot A[n-4] \\
 &\quad + A[n-2] \cdot \overline{A[n-3]}) \cdot \overline{A[n-1]} \\
 &\vdots \\
 A_r[i] &= (\overline{A[i]} \cdot A[i-1] \cdot A[i-2] + A[i] \cdot \overline{A[i-1]}) \cdot \prod_{i=i+1}^{n-1} \overline{A[i]} \\
 &\vdots \\
 A_r[3] &= (\overline{A[3]} \cdot A[2] \cdot A[1] + A[3] \cdot \overline{A[2]}) \cdot \prod_{i=4}^{n-1} \overline{A[i]} \\
 A_r[2] &= A[2] \cdot \overline{A[1]} \cdot \prod_{i=3}^{n-1} \overline{A[i]} \\
 A_r[1] &= A[1] \cdot \prod_{i=2}^{n-1} \overline{A[i]} \\
 A_r[0] &= A[0] \cdot \prod_{i=1}^{n-1} \overline{A[i]} \tag{3}
 \end{aligned}$$

The output of this adder and therefore the results of $A_r \times B_r$ are the inputs of the subtractor block whose output is that the definite quantity of the output of the proposed multiplier as a result of A_r and B_r are within the style of $2n$, the inputs of the subtractor might take one amongst the 3 input patterns. The corresponding output patterns are shown. The kinds of the inputs and output influence us to conceive a fundamental circuit supported the subsequent expression:

$$\text{Out} = (P \text{ XOR } Z) \text{ AND } ((P \ll 1) \text{ XOR } (P \text{ XOR } Z)) \text{ or } ((P \text{ AND } Z) \ll 1) \tag{4}$$

Where P is $A_r \times B + B_r \times A$ and Z is $A_r \times B_r$. The corresponding circuit for implementing this expression is smaller and quicker than the standard subtraction circuit. Finally, if the sign of the final multiplication result ought to be negative, the output of the subtractor is negated within the sign set block. To negate values, that have the two ought to complement illustration, the corresponding circuit supported $\sim X + 1$ to be used. To extend the speed of negation operation,

one could skip the incrimination method within the negating section by acceptant to its associated error. As are seen later, the significance of the error decreases because the input widths will increase. In this paper, if the negation is performed precisely (approximately), the implementation is named signed Roba (Roba number [approximate S-Roba (AS-Roba) multiplier]). Within the case wherever the inputs are forever

positive, to extend the speed and lower the consumption of power, the sign detecto, and sign set blocks are omitted from the design, providing us with the design referred to as unsigned ROBA (U-ROBA) number. during this case, the output dimension of the rounding block is $n+1$ wherever this bit is decided to depend on $Ar[n]=A[n-1] \cdot A[n-2]$. This can be as a result of within the case of unsigned $11x \dots x$ (where x denotes doesn't care) with the bit dimension of n , its rounding value is $10\dots 0$ with the bit dimension of $n + 1$. Therefore, the input bit dimension of the shifters is $n + 1$. However, as a result of the maximal quantity of shifting is $n - 1$, $2n$ is taken into account for the output bit dimension of the shifters.

The accuracy of ROBA Multiplier in this section

Inaccuracies of the three architectures mentioned are considerable. The inaccuracies of the U-Roba number and S-Roba number, that originate from omitting the term $(Ar-A) \times (Br-B)$ from the correct multiplication of $A \times B$, are identical. Hence, the error is

$$\text{error}(A, B) = \frac{(A_r - A)(B_r - B)}{AB} \quad (5)$$

Assuming Ar and Br are equating to 2^n and 2^m , identically, the most error happens once A and B are equating to 3×2^n and 3×2^m , severally. In this case, each Ar and Br has the most arithmetic distinction from their corresponding inputs. Thus

$$\max\{\text{error}(A, B)\} = \frac{(2^n - 3 \times 2^{n-2})(2^m - 3 \times 2^{m-2})}{(3 \times 2^{n-2}) \times (3 \times 2^{m-2})} = \frac{1}{9} \quad (6)$$

In the case of the AS-Roba multiplier factor, the error includes a term because of the approximate negation (approximate negation). Therefore, within the worst case (where each input square measure negative), one could acquire the most error form.

$$\text{error}(A, B) = \frac{(\bar{A}_r - \bar{A})(\bar{B}_r - \bar{B})}{AB} + \frac{\bar{A} + \bar{B} + 1}{AB} \quad (7)$$

Compared with (5), the second term comes from the negation approximation obtained from the following relation:

$$A \times B = (\bar{A}+1)(\bar{B}+1) = \bar{A} + \bar{B} + 1 + \bar{A} \times \bar{B} \approx \bar{A} \times \bar{B} \quad (8)$$

Hence, within the case wherever a minimum of one amongst the inputs is negative, the AS-Roba number error is larger than that of the two other Roba multiplier types. Also, once each of the inputs area unit negative, though the final result is going to be positive, one still has to negate the negative inputs. Depending

on this formulation, one amongst the inputs is -1 , the most error that is 100 %, occurs. To minimize the most error of this case, one could use a detector to detect the case whenever one amongst the inputs is -1 , and bypass the multiplication method and generate the output by negating the opposite input. It's clear that this solution has some delay and power consumption overhead. Additionally to the most error, the incidence rate of an error condition is obtained because of the magnitude relation of the maximum error occurrences to the overall number of outputs. This error rate is another accuracy measure parameter. Here, all the input combinations are assumed to occur. Within the case of n-bit U-ROBA multiplier, there are $n - 1$ cases for every input wherever the rounded worth has the most distinction to the particular number. The error happens once these numbers are the input operands. This corresponds to $(n - 1)2$ cases. Within the case of S-ROBA multiplier, for every operand, there are $2(n-2)$ cases wherever the rounded operands have the most error. Hence, the same as the U ROBA multiplier, most the error happens once each of the rounded operands has the error that creates the quantity of maximum error incidence adequate $(2(n - 2))2$. Finally, within the case of the AS-ROBA multiplier, as mentioned before, the most error happens once one amongst the inputs is -1 . Hence, the quantity of most error occurrences can be equal to $2 \times 2^{n-1} - 1$ ($2^n - 1$). On the other hand, in the cases of the U- Roba and Roba multipliers when the absolute value of the input operand of the multiplier is within the variety of the 2^m , the output results of the Roba multiplier is exactly noticed.

Hence, the numbers of correct outputs within the cases of the U-Roba multiplier factor and S-Roba multipliers are $2^{n+1}2^{n-(n+1)2}$ and $n2^{n+2}-4n2$, respectively. Within the case of the AS-Roba multiplier factor, when both inputs are positive, the multiplier behaves kind of like the other 2 Roba multiplier factor architectures, and hence, when one of the inputs is within the variety of 2^m , the output is an exact one. In addition, there are other combination's that results in the correct output. One example of such cases is $(A-AR) (\bar{B} - \bar{B} R) + A = 1$. Analytically finding all the combination's with correct (exact) output is too difficult, and hence for the AS-Roba multiplier, we have a tendency to use the boundary of the correct output range that's equal to $n2^n - n2$.

RESULTS

Proposed System Results:



Figure 2: RTL schematic of the proposed system

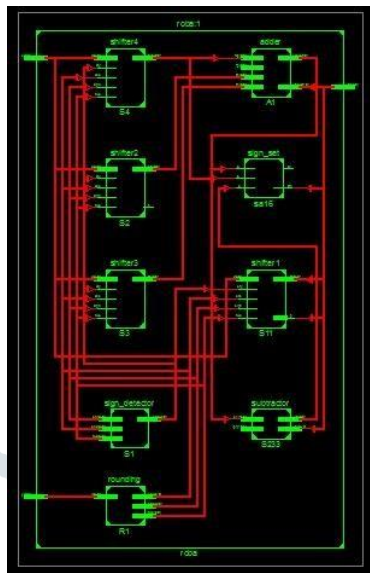


Figure 3: Technology schematic of proposed system

Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of 4 input LUTs	156	9,312	1%	
Number of occupied Slices	88	4,656	1%	
Number of Slices containing only related logic	88	88	100%	
Number of Slices containing unrelated logic	0	88	0%	
Total Number of 4 input LUTs	156	9,312	1%	
Number of bonded IOBs	32	232	13%	
Average Fanout of Non-Clock Nets	4.16			

Figure 4: area

```

-----
Total                21.412ns (13.455ns logic, 7.957ns route)
                    (62.8% logic, 37.2% route)
=====

Total REAL time to Xst completion: 19.00 secs
Total CPU time to Xst completion: 19.02 secs
    
```

Figure 4: delay

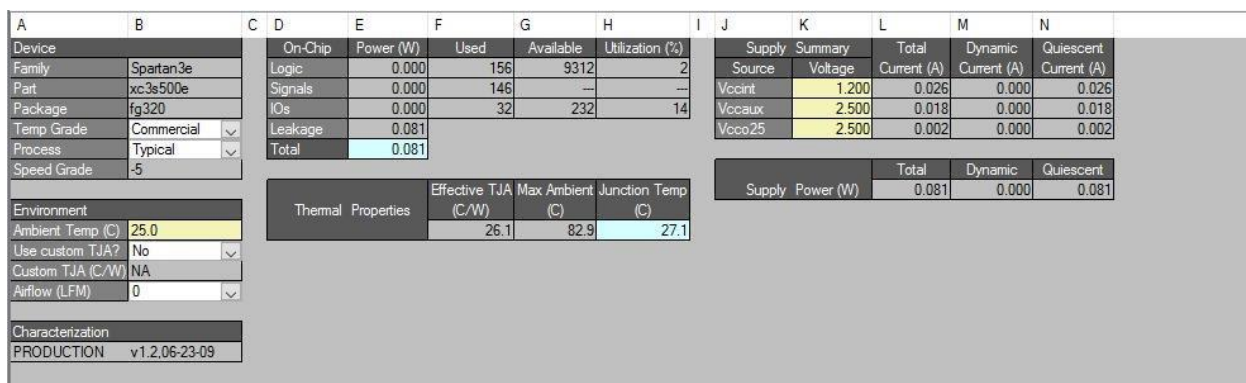


Figure 4: power

Table 1. Comparison Table

Parameter	Existing System	Proposed System
Delay(ns)	38.26	23.94
No. of LUTs	437	112
No. of I/o s	163	145

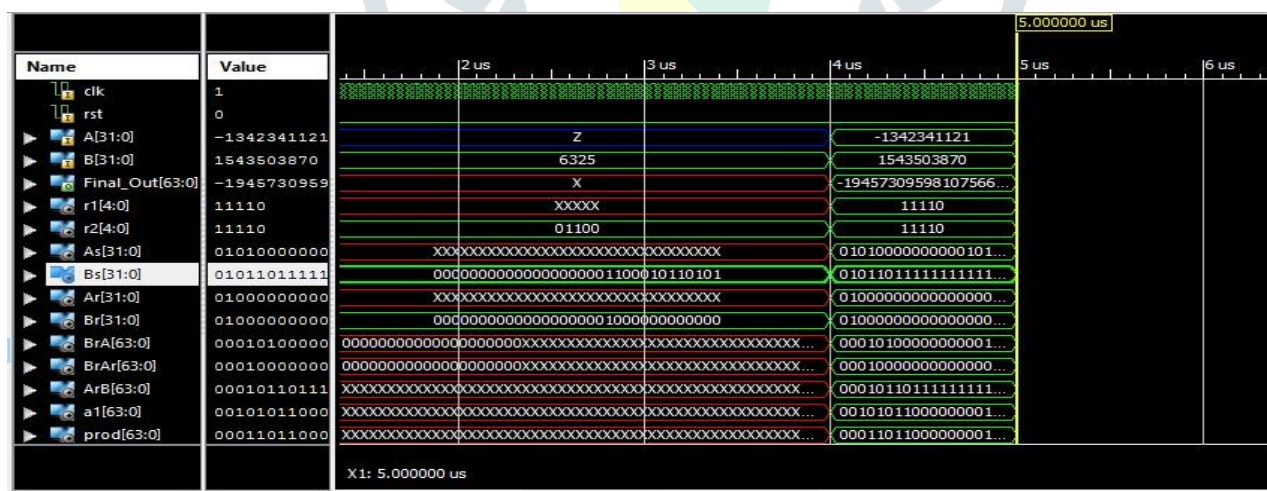
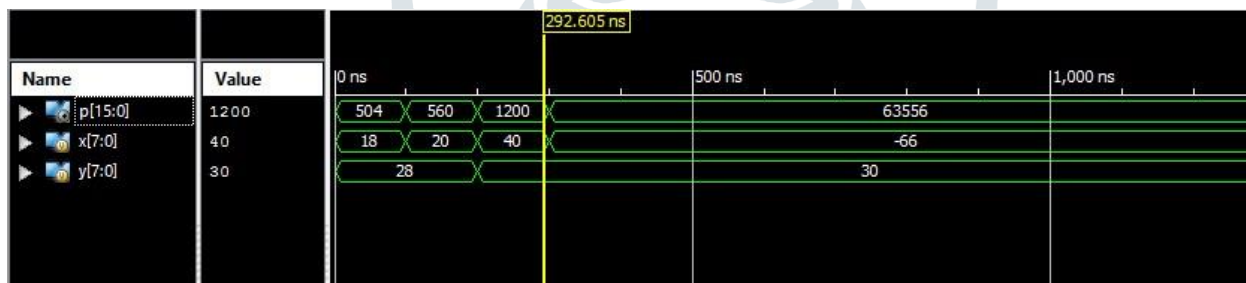


Figure 5: signed and unsigned simulation results

CONCLUSION

In this paper, we tend to plan a high speed and energy economical approximate multiplier factor referred to as Roba multiplier factor. The proposed multiplier that had high accuracy was depending on the rounding error of the inputs in 2n. During this approach, the process intensive a part of the multiplication was omitted

rising speed and energy consumption at the worth of a tiny low error. The proposed approach was applicable to each signed and unsigned multiplications. Three hardware implementations of the approximate multiplier including both one for the unsigned and two for the signed operations were mentioned. The efficiencies of the proposed multipliers were evaluated by equivalent to those of some correct and approximate multipliers using totally different style parameters. The results discovered that, in most (all) cases, the Roba multiplier architectures outperformed the corresponding approximate (exact) multipliers. Also, the efficiency of the proposed approximate multiplication approach was studied in two image process applications of sharpening and smoothing. The comparison discovered identical image qualities as those of accurate multiplication algorithms.

References

- Dola Sanjay S, Prof. S Varadarajan” Finite-Difference Time-Domain”, National Conference, Jan. Pg-60, 2011.
- Dola Sanjay S, Prof. S Varadarajan” Radiated Emission Measurement of Microstrip Patch Antenna”, National conference on Thrust areas in Engineering,Jan.2012.
- Dola Sanjay S, B Kalyan Kumar, Prof. S Varadarajan” A moment method solution for spherical structures used in shielding electronic components”, Applied Mathematical Science, Vol.6,2012, no.7,333-341.
- Dola Sanjay S, Prof. S Varadarajan” AMSA as EMI/EMC Sensor”, Journal of innovation in Electronics and Communication-Special Issue, Vol. 2, issue 2,ISSN:2249-9946, Jan.2012, 126-129.
- Dola Sanjay S, Prof. S Varadarajan” Implementation of Conducted Emission and Conducted Susceptibility of Pressure Sensor”, Proceedings of ICIECE-2012, ISSN: 2249-9946, Jul.2012, 38.
- Dola Sanjay S, Prof. S Varadarajan, B Kalyan Kumar,” Implementation of Conducted Emission and Conducted Susceptibility of Pressure Sensor”, IJSER,Vol.4,Jan.2013.
- Dola Sanjay S, Prof. S Varadarajan, B Kalyan Kumar,” Analysis of Conducted Emission and Susceptibility, Radiated Emission and Susceptibility due to Stereo Board”, NCETEC, Apr.2013, 108-112.
- Dola Sanjay S, Prof. S Varadarajan, “Reduction of Electromagnetic Interference Using Micro-strip Filter”, ICPVS 2014, Elsevier Publications, ISBN:978-93-5107-228-7, Mar-2014,90-93.
- R Venkata Subbamma, Dola Sanjay S,” Optimization of Delay and Error Correcting using Fault Tolerant Parallel Filters, IJVDCS, ISSN 2322-0929, Vol 04, Issue 05, May 2016, Pages:0337-0341.

- Kancharla Anusha Rani, Dola Sanjay S, Yedukondalu Udara, “Efficient Scan Based Testing for Memories”, IJSETR,ISSN 2319-8885, Vol.06, Issue 08, Feb-2017, pages: 1655-1657.
- Madhu babu Kodele, Dr.Dola Sanjay S, S Swapna” A study of yogic practice effects on respiratory system of human body” IJHPECSS, ISSN 2231-3265, Vol 26, Issue 3, IF 3.565, pages: 60-65.
- Dr.Dola SanjayS, P. Geetha Lavanya, P.Jagapathi Raju, M.Sai Kishore, T.N.V.Krishna Priya “ Denoising the Spectral Information of Non–Stationary Image using DWT” IJECS, ISSN 2348-117X, Vol 6, Issue 7 July 2017,pages: 594-598.
- Prof. Dola Sanjay S, G. Pallavi , B. Tarun Kumar, M. Tejaswini , M. Ratna Kireeti “Redundancy Elimination of Stationary Image Using DWT” IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 12, Issue 3, Ver. II (May - June 2017), PP 72-79,10.9790/2834-1203027279(DOI).
- Madhu babu Kodele, Dr. Dola Sanjay S, S Swapna, D V Lakshmana Rao ” Hot Spot Technology in the Sport of Kho-Kho A study of yogic practice effects on respiratory system of human body” IJHPECSS, ISSN 2231-3265, Vol 27, Issue 2, IF 3.565, pages: 294-296.
- Dr. Udara Yedukondalu, Sasi Priya Musunuri , Dr. Dola Sanjay. S “VLSI Design of An Area Efficient Architecture of DSP Accelerator Using Dadda Algorithm” IJIEMR, Vol 06 Issue 08, Sept 2017 ISSN 2456 – 5083 Page 327-333.
- Kancharla Anusha Rani, Dola Sanjay S, Yedukondalu Udara, “A MEMORY TESTING USING REBIST TECHNIQUE”,IJIEMR, Volume 06, Issue 04, June 2017 ISSN: 2456 - 5083 Page 1431-1435.
- Dr. Dola Sanjay.S, Ch.Lakshmi, G.Jayachandra, G. Divya Rupini ,K. Ratnakumar, V Kalpana” Cash less Toll System Using Rppide” *International Journal of Engineering Science Invention (IJESI) ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726,Volume 7 Issue 4 Ver. II, April 2018, PP 29-33.*
- Dr. Dola Sanjay.S, Sri Durga K, Jayasri M, Mounika S, Eswar Chandu Y , Kumara Varma M” Thermmatology Scan Methodology Using Patel Sensor and Roku Processor” *International Journal of Engineering Science Invention (IJESI) ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726,Volume 7 Issue 4 Ver. II, April 2018, PP 34-38.*
- K Sai Rohith, M Vishnu Vardhan, B G Rahul, Dola Sanjay S “Pavement Design Using Benkelman Beam Method” International Journal Of Scientific & Engineering Research, Volume 8, Issue 1, January-2017 ISSN 2229-5518 PP 1201-1206.

- *Nishnanth.Dulam, Ganesh.Burila, B G Rahul, Dola Sanjay S*” Road Accidents Study Based on Regression Model: a Case Study of Vijayawada city to Hanuman Junction” International Journal of Scientific & Engineering Research Volume 9, Issue 5, May-2018 1207 ISSN 2229-5518 PP 1207-1218
- Dr. Dola Sanjay S, P K Ratnam “Reduction of power in confined field multiplier using sorting technique”, International Journal of Research e- Volume 05 Issue 16, ISSN: 2348-6848 p-ISSN: 2348-795X PP 712-715 June 2018
- Dr. Dola Sanjay S, P K Ratnam “Reduction of power in confined field multiplier using sorting technique”,
2018 IJCRT | Volume 6, Issue 2 April 2018 | ISSN: 2320-2882 PP 798-801 June 2018
- Dr. Dola Sanjay S, “Implementation of Low Power Transposed FIR Filter using Clock Gating Technique”
2018 IJSRR | Volume 7, Issue 11 Nov 2018 | ISSN: 2279-543X, DOI:16.10089/IJSRR.
- B S Satish, Dola Sanjay S, A Ranganayakulu, S Jagan Mohan, Ganesan P “Advanced Design of Service Robot for Aged and Handicapped Using Rasberry Pi” 2019 Springer CCIS 922, https://doi.org/10.1007/978-981-10-8660-1_74, pp854-864, 2019, ISSN: 1865-0929.
- B S Satish, Dola Sanjay S, A Ranganayakulu, S Jagan Mohan, Ganesan P “An Approach to Threshold based Human Skin Color Recognition and Segmentation in Different Color Models” 2019 Springer CCIS 922, https://doi.org/10.1007/978-981-10-8660-1_74, pp 920-931, 2019, ISSN: 1865-0929.
- KHASIM SHAIK, Dr. Dola Sanjay S, Dr. Anupama A ” An Approach For Artificial Intelligence Techniques For Combating Cyber Crimes” Universal Review, Volume VIII, Issue III, MARCH/2019, pp 597-601, ISSN NO : 2277-2723