



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJIRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Special Issue 1, April 2022

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.118

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

Design of High Performance and Low Power Rounding Based Approximate Multiplier by Using HDL

V Venkanna⁽¹⁾, Dr. N Ashok Kumar⁽²⁾

Research Scholar, Dept. of ECE, Kasireddy Narayanreddy College of Engineering & Research, JNTUH, Telangana, India⁽¹⁾

Assistant Professor, Head of the Department ECE, Kasireddy Narayanreddy College of Engineering & Research, JNTUH, Telangana, India^{(2),(3)}

ABSTRACT: In this paper, we have a tendency to propose associate degree approximate number that's high speed however energy economical. The approach is to round the operands to the closest exponent of 2. This fashion the machine intensive a part of the multiplication is omitted rising speed and energy consumption at the price of a tiny low error. The projected approach is applicable to each signed and unsigned multiplications. We have a tendency to propose 3 hardware implementations of the approximate number that includes one for the unsigned and 2 for the signed operations. The potency of the projected multiplier is evaluated by comparison its performance with those of some approximate and accurate multipliers victimization completely different style parameters. Additionally, the effectiveness of the projected approximate number is studied in 2 image process applications, i.e., image sharpening and smoothing. The projected design of this paper analysis the logic size, space and power consumption victimization Xilinx 14.5.

KEYWORDS: Accuracy, approximate computing, energy efficient, error analysis, high speed, multiplier.

I. INTRODUCTION

Some of the previous works in the field of approximate multipliers are briefly reviewed. In an approximate multiplier and an approximate adder based on a technique named broken array multiplier (BAM) were proposed. By applying the BAM approximation method to the conventional modified Booth multiplier, an approximate signed Booth multiplier was presented. The approximate multiplier provided power consumption savings form 28% to 58.6% and area reductions from 19.7% to 41.8% for different word lengths in comparison with a regular Booth multiplier. Kulkarni et al. suggested an approximate multiplier consisting of a number of 2x2 inaccurate building blocks that saved the power by 31.8%–45.4% over an accurate multiplier. An approximate signed 32-bit multiplier for speculation purposes in pipelined processors was designed. It was 20% faster than a full-adder-based tree multiplier while having a probability of error of around 14%. An error-tolerant multiplier, which computed the approximate result by dividing the multiplication into one accurate and one approximate part, was introduced, in which the accuracies for different bit widths were reported. In the case of a 12-bit multiplier, a power saving of more than 50% was reported. In two approximate 4:2 compressors for utilizing in a regular Dadda multiplier were designed and analyzed. The use of approximate multipliers in image processing applications, which leads to reductions in power consumption, delay, and transistor count compared with those of an exact multiplier design, has

been discussed in the literature. In, an accuracy-configurable multiplier architecture (ACMA) was suggested for error resilient systems. To increase its throughput, the ACMA made use of a technique called carry-in prediction that worked based on a pre-computation logic. When compared with the exact one, the proposed approximate multiplication resulted in nearly 50% reduction in the latency by reducing the critical path. Also, Bhardwaj et al. presented an approximate Wallace tree multiplier (AWTM). Again; it invoked the carry-in prediction to reduce the critical path. In this work, AWTM was used in a real-time benchmark image application showing about 40% and 30% reductions in the power and area, respectively, without any image quality loss compared with the case of using an accurate Wallace tree multiplier (WTM) structure.

II. LITERATURE REVIEW

R. Hegde and N. R. Shan hag in 2001 proposed a framework for energy efficient digital signal processing [15]. The supply voltage is scaled further, which is the critical voltage required to fix the critical path delay to the throughput. This introduction of input-dependent errors leads to degrade in the algorithmic performance, which is compensated through the Algorithmic Noise Tolerance (ANT) schemes. A prediction based error-control scheme is proposed to the filter algorithm performance and the presence of errors computations. The CMOS technology reduction in energy dissipation has made possible due to energy-efficient design techniques at all possible levels of design hierarchy. V. K. Chippa et.al.in 2010 proposed scalable effort hardware design as an approach to tap the reservoir of algorithmic recover and translate it into highly efficient hardware implementations [14]. The basic idea of the scalable effort design is to identify mechanisms at each level of design abstraction like circuit, architecture and algorithm that can be used to varying the computational effort. A second major idea of the scalable effort design approach is that fully based on the potential of algorithmic recover requires synergistic cross layer optimization of scaling mechanisms are identified at different levels of design abstraction. C. H. Chang and R. K. Satzoda in 2010 proposed to ignore the least significant columns in the partial product (pp) bit matrix obtained from an n-bit multiplication [13]. A small amount of additional hardware is added to compensate for the truncation errors. This method achieves lower average and spread of errors by means of adaptive pseudo carry compensation and a constant bias that is independent of supply. By especially the symmetry of the multiplexer-based array multiplier the partial product bits are generated by the multiplexers. The truncated multiplier can be assembling in a carry-save adder format to reduce the area and improve the speed over than other truncated array multipliers.

III. EXISTING METHODOLOGY

The Sir Bernard Law solution for modular duplication is known as the fastest series of rules to measure $xy \bmod n$ in computer systems when the values of x , y and n are massive. Within this lecture, we will define the Montgomery series of modular multiplication policies. This is one of the algorithm information that my college students had difficulty apprehending.

Montgomery Multiplication is a method for wearing lots quicker compact replicate of major cryptographic sets. Bernard Law Sir Bernard Law arithmetic is a compact replica calculator in which a , b and moreover n declare integers[2] Bernard Law Sir Bernard Law gadgets turns proper numbers right into a website named 1st viscount Montgomery of alamein region as well as includes critical operations and returned converts which eliminates the need of more division t The steps concerned about Sir Bernard Law Multiplication are:

1. Consider two integers a and b less than modulo n
2. Introduce a number R greater than n such that gcd (R, N) =1
3. Find two integers R-1 and N-1 such that: $RR^{-1} - NN^{-1} = 1$.
4. Transform the multipliers to Montgomery space by the following multiplication and reduction:

$$A = a \times R \text{ mod } N$$

$$B = b \times R \text{ mod } N.$$

5. Compute the Montgomery multiplication using the algorithm in Figure 1 which computes product of A and B giving the result S in Montgomery space that is

$$S = A \times B \times R^{-1} \text{ mod } N$$

6. The final result can be obtained by back transforming from Montgomery space:

$$s = S \times R^{-1} \text{ mod } N.$$

The rationale behind this computation is to find a proper value of q so that, at the k-th iteration, the value of $A + xkY + qN$ is a multiple of β . and that this value is bounded by $(R + \beta k)N$. As explained above, the value of q is

$$q = (A + xkY)N^{-1} \text{ mod } \beta,$$

Montgomery Algorithm:**Function**

montgomery(X, Y)

Begin

A := 0;

For k := 0 to m - 1 do begin

q := (a0 + xky0)(β - n0) - 1 β

mod β; A := A + xkY + qN;

A := A/β; end;

Return A;**End.**

In this process, the high-priced branch machine typically used for compact bargaining is modified by way of clean trade methods by translating the operands into the name of the RNS area prior to service and retransforming the result after technique. A radix R is selected to be 2 extra than one word size and also extra than the node, i.e. $R = 2w$ M. To be relatively immoderate, the collection of rules to include R identical to M, i.e. should have no odd non-trivial divisors. With R third power, it's easily pleased to settle on an odd board. This moreover fits properly with the cryptographic scheme we're concentrating on, in which the eigenvalues is each a top normally nonsensical but 2or the element of two primes and, moreover, because of this weird as cool. RNS integer representations are pointed to as M compounds and are typically denominated as the integer component calls a bar over it. An integer a is properly modified into its equivalent M-residue by increasing it through R and also rising M-module. The re-transformation is done in another smooth design by separating the residue through R modulo M.

Everything framework supplied in the previous region is using Verilog HDL and synthesized using Synopsys Software compiler. Implementation details on simulation results occurs in this portion.

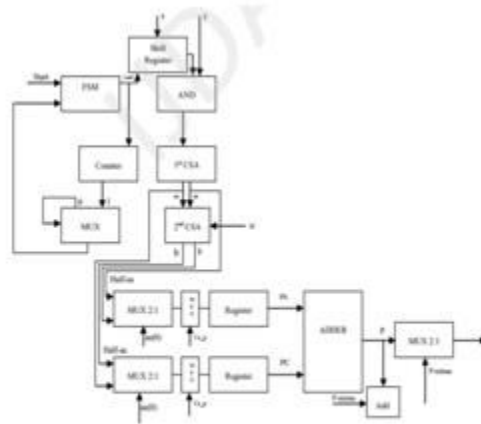


Fig.3.1 Proposed system architecture.

Testing a complicated electronic interface math pattern poses a particular challenge to the exam bench clothier. Obviously, the research types to be generated can not be purely random variables, however they need considerable expertise in computing this. Consequently, next to join, take a good look at types, the corresponding final results variables for assessment.

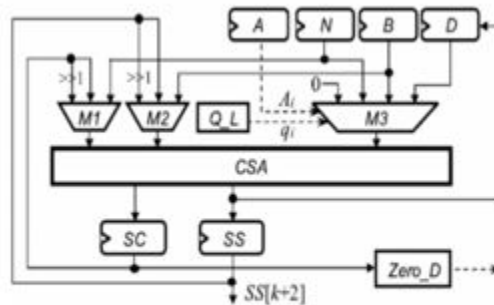


Fig. 3.2 Diagram of Montgomery Modular Multiplier.

Along with the right really worth the details, understanding the clock period when it reaches or exits the circuit is also utterly crucial. The framework involves millions of phases of logic or intermediary signs on rates; as a result, the end result of tests no longer correlates consistently with the exam vector middle. Useful testing does not find the only major path delays a model has. The critical path survival at once relates the first-rate frequency of the clock a system should maintain which would be identified at the point of synthesis.

Disadvantages:

- Not provided the output with Chrominance
- More Area, and Power
- Low Performance

PROPOSED SYSTEM:

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

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

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:

$$A * B = (A_r * B) + (B_r * A) - (A_r * B_r) \quad (2)$$

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 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 ROBA multiplier 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 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 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.

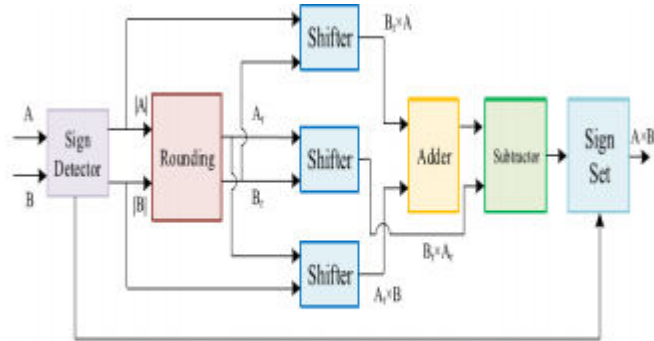


Fig 1. Block diagram for the hardware implementation of the proposed multiplier.

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

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

Hardware Implementation of RoBA Multiplier: Based on (2), 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 $2n$. 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: In the proposed equation, $A_r[i]$ is one in two cases. In the first case, $A[i]$ is one and all the bits on its left side are zero while $A[i-1]$ is zero. In the second case, when $A[i]$ and all its left-side bits are zero, $A[i-1]$ and $A[i-2]$ are both one. Having determined the rounding values, using three barrel shifter blocks,

the products $A_r \times B_r$, $A_r \times B$, and $B_r \times A$ are calculated. Hence, the amount of shifting is determined based on $\log_2 A_r - 1$ (or $\log_2 B_r - 1$) in the case of A (or B) operand. Here, the input bit width of the shifter blocks is n , while their outputs are $2n$. A single 2nbit Kogge-Stone adder is used to calculate the summation of $A_r \times B$ and $B_r \times A$. The output of this adder and the result of $A_r \times B_r$ are the inputs of the subtract or block whose output is the absolute value of the output of the proposed multiplier. Because A_r and B_r are in the form of $2n$, the inputs of the subtract or may take one of the three input patterns shown in Table I. The corresponding output patterns are also shown in Table I. The forms of the inputs and output inspired us to conceive a simple circuit based on the following expression:

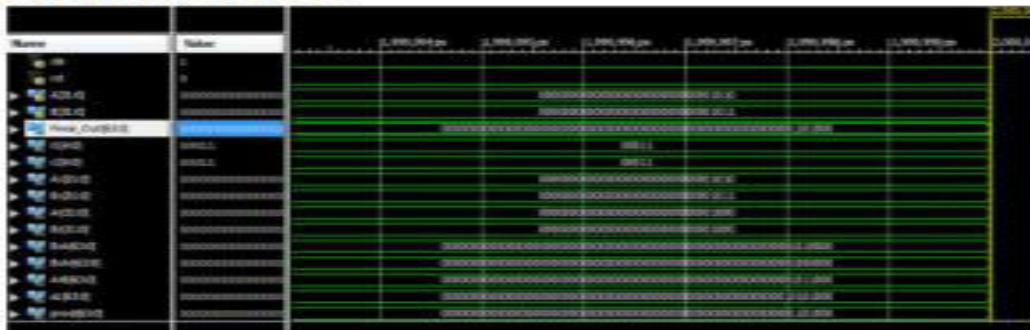
$$\text{Out} = (P \oplus Z) \text{ AND } (\{(P \ll 1) \oplus (P \oplus Z)\} \text{ or } \{(P \text{ AND } Z) \ll 1\}) \tag{4}$$

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 faster than the conventional subtraction circuit.

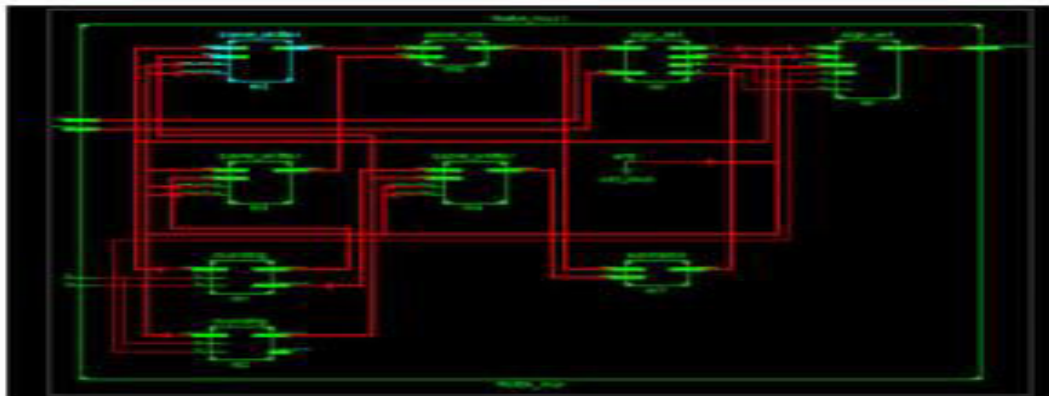
IV. RESULTS

Different blocks of proposed system are designed coded in VERILOG HDL, simulated in I simulator and Xilinx ISE is the software tool used for FPGA synthesis.

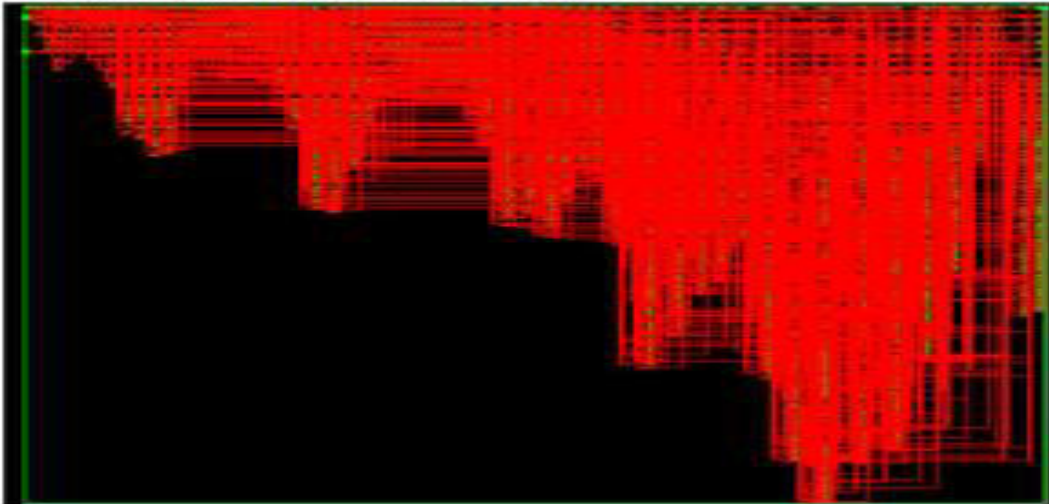
SIMULATION:



RTL SCHEMATIC:



TECHNOLOGY SCHEMATIC:



DESIGN SUMMARY:

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	2256	4636	49%
Number of Slice Flip-Flops	424	9312	4%
Number of 4-input LUTs	4007	9312	43%
Number of bonded I/Os	226	232	97%
Number of GCLs	1	24	4%

TIMING REPORT:

Timing Summary:

Speed Grade: -5

Minimum period: 19.790ns (Maximum Frequency: 50.530MHz)
 Minimum input arrival time before clock: 21.539ns
 Maximum output required time after clock: 24.403ns
 Maximum combinational path delay: 24.194ns

V. CONCLUSION

In this paper, we 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 $2n$. 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.

REFERENCES

- [1] 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.
- [2] V. Gupta, D. Mohapatra, A. Raghunathan, and K. Roy, "Low-power digital signal processing using approximate adders," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.
- [3] 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.
- [4] 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.
- [5] F. Farshchi, M. S. Abrishami, and S. M. Fakhraie, "New approximate multiplier for low power digital signal processing," in Proc. 17th Int. Symp. Comput. Archit. Digit. Syst. (CADSD), Oct. 2013, pp. 25–30.
- [6] P. Kulkarni, P. Gupta, and M. Ercegovic, "Trading accuracy for power with an underdesigned multiplier architecture," in Proc. 24th Int. Conf. VLSI Design, Jan. 2011, pp. 346–351.
- [7] D. R. Kelly, B. J. Phillips, and S. Al-Sarawi, "Approximate signed binary integer multipliers for arithmetic data value speculation," in Proc. Conf. Design Archit. Signal Image Process., 2009, pp. 97–104. [8] K. Y. Kyaw, W. L. Goh, and K. S. Yeo, "Lowpower high-speed multiplier for error-tolerant application," in Proc. IEEE Int. Conf. Electron Devices Solid-State Circuits (EDSSC), Dec. 2010, pp. 1–4.
- [9] A. Momeni, J. Han, P. Montuschi, and F. Lombardi, "Design and analysis of approximate compressors for multiplication," IEEE Trans. Comput., vol. 64, no. 4, pp. 984–994, Apr. 2015.
- [10] K. Bhardwaj and P. S. Mane, "ACMA: Accuracy-configurable multiplier architecture for error-resilient system-on-chip," in Proc. 8th Int. Workshop Reconfigurable Commun.-Centric Syst.-Chip, 2013, pp. 1–6.

BIOGRAPHY

1. V VENKANNA



He is a Research Scholar from Sri Satya Sai University of Technology and Medical Sciences, one of the most prestigious universities India. He has completed his Post Graduation from Avanthi Group of Institutions, and Under Graduation from Pulipati Prasad Institute of Technology & Science, JNTU Hyderabad. He was a faculty of electronics in Brilliant Group of Technical Institutions. He has guided many projects at UG level. His fields of interest are VLSI Design, Embedded Systems.

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization

Volume 11, Special Issue 1, April 2022

9th National Conference on Frontiers in Communication and Signal Processing Systems (NCFCSPS '22)

28th -29th April 2022

Organized by

Department of ECE, Adhiyamaan College of Engineering, Hosur, Tamilnadu, India

2.Dr. N ASHOK KUMAR



Dr. Ashok Kumar Nallagonda is working as Associate Professor & Head of the Department in Kasireddy Narayanreddy College of Engineering & Research affiliated under JNTU, Hyderabad, India. He served as an Assistant Professor from 2008 to 2017 in AVANTHI GROUP OF INSTITUTIONS. He has received his B.Tech degree in Electronics and Communication Engineering in 2005 from Acharya Nagarjuna University, Guntur, and Andhra Pradesh, India. He received M. Tech Degree in Radar & microwave Engineering in 2008 and Ph. D. in CMOS IMAGE PROCESSING in 2017 from DELHI TECHNOLOGICAL UNIVERSITY DELHI, India. His research interests include VLSI, Image Processing,

Antennas, Radar Systems, Microwave Engineering As of today, he has published more research papers in various international conferences, journals, and book chapters. He is also served as a reviewer for several conferences.



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118



ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



www.ijirset.com

Scan to save the contact details