iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 Issue: VI Month of publication: June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35052

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VI Jun 2021- Available at www.ijraset.com

Design and Implementation of a 32-bit Floating Point Unit

Kishan Maladkar[1], H V Ravish Aradhya[2]
[1, 2]Dept. of Electronics and Communications, R V College of Engineering

Abstract: A Floating Point Unit is a math co-processor that is in the most demand of Digital Signal Processing (DSP), Processors and more. It is used to perform functions or operations on floating point numbers like addition, subtraction, multiplication, division, square root and more. It is specifically designed to carry out mathematical operations and it can be emulated in CPU. Floating point unit is a common operation used in advanced Digital Signal Processing and various processor applications. The aim was to develop an optimized floating point unit so that the delay was reduced and efficiency was increased. The floating point unit has been written according to IEEE 754 standard and the entire design has been coded in Verilog HDL. The results are improved by 12% with the usage of Vedic multiplier that is a delay of 4.450ns as compared to 5.123ns with an array multiplier. Designs can be further optimized using low power designing techniques at architectural level. Different behaviour can be observed for different size and technologies.
Keywords: Floating Point Unit, co-processor, Digital Signal Processing, IEEE 754, Verilog.

I. INTRODUCTION

Digital Signal Processors (DSPs) or any processors that involve complex operations like multiplication and/or accumulation operations with high precision as a major portion and Floating Point Unit plays a crucial role in implementing it, especially in high performance DSPs. Floating Point is also widely known as math co-processor that is used to operate number quickly with more accuracy than the basic processor.

A floating point unit contains digit sequence in three parts that is sign, mantissa and exponent. The sign can be plus or minus, mantissa is sequence of digits and exponent is the power of magnitude. The main operation of a floating point unit includes addition, subtraction, multiplication, division and square root.

Floating point unit can be single precision or double precision. A single precision consists of 32 bits and double precision consists of 64 bits. Fig 1 shows a 32 bit floating point unit with 1 sign bit, 8 bit of exponent and mantissa of 23 bits. Fig 2 shows a 64 bit floating point unit with 1 sign bit same as single precision and 11 bit of exponent and 52 bits of mantissa.

Sign (1-bit)	Exponent (8 bit)	Mantissa (23 bits)

Fig. 1 Single Precision 32 bit

Sign (1-bit)	Exponent (11 bit)	Mantissa (52 bits)

Fig. 2 Double Precision 64 bit

In paper [1], improved timing of division iteration with the help of divisor pre-scaling technique can be seen. Here, the operation is carried out in parallel to save time consumption compute faster and increase efficiency.

In paper [2], the proposed design computes a four term dot product in a single unit to achieve higher accuracy and performance from the traditional method. The design requires complex processing like rounding, normalization that increases the power consumption and area.

In paper [3], the arithmetic operations are executed with minimum delay or within a single clock. A smaller chip will be required. The designed block does not require flip flops it requires only combinatorial blocks that will be executed with minimum delay that results into faster computation.

In paper [4], a 32 bit floating point multiplier has been developed using an array multiplier along with a modified full adder to increase the speed of the operation, the multiplier generates only required MSB bits. The designed pipelined architecture reduces time consumption and increases efficiency.

In paper [5], a low power and area efficient floating point four term fused dot product that is used in Vedic mathematics. The power consumption and LUT's are reduced with this method.

In paper [6], the floating point unit can perform subtraction, addition that operates on double precision floating point numbers. The design helps in faster computation, less delay and occupies less area.

In paper [7], a IEEE 754 compliant floating point unit is developed for addition, subtraction that supports both 32 bit and 64 bit operands. This design helps in faster computation, lesser delay.

In paper [8], a single precision floating point unit are used for addition and subtraction with two pre-normalization units are used and post-normalization is used for mantissa part.

In paper [9], a 24 bit Vedic multiplier with carry save adder has been developed, it calculates mantissa part in single precision and it outperforms others in terms of speed, and delay.

In paper [10], design and implementation of 32 bit floating point for DSP application has been used, a 332 bit DSP processor and MAC unit are used with IEEE 754, 32 bit floating point unit.

In paper [11], a Vedic Multiplier is one of the efficient multipliers to decrease the delay and improve the performance. It gives a better performance when compared with that of other floating point unit.

The motivation to take up this project is to develop a high speed floating-point computation that is essential for a large class of problems, like computer modeling and simulation, computer graphics, image processing, hydrodynamics, and computer-aided design. And to develop a fast and efficient floating point unit that can be used for various DSP applications and processors such that the computation time is reduced also to improve the accuracy of the computational device.

II. FLOATING POINT UNIT

Floating Point Unit is a math co-processor specifically designed for operation on floating point numbers that can handle operations like addition, subtraction, multiplication, division and more. There are three ways to carry out operation, a floating point unit can be emulator that is a floating point library. Emulators can save extra hardware costs but they are slow. Second, it can use add on floating point unit to a CPU to speed up math operations. In single precision format the exponent is of 8 bit wide and has a range of -127 to 128. In double-precision format the exponent has 11 bit wide range of -1023 to 1024.

The table 1 shows the difference between single precision and double precision floating point unit in detail. The advantage of using floating point, it gives wider range of values in comparison to fixed format. Another advantage is it has flexibility and high accuracy or precision that helps complex problems. Floating point unit can be used in audio and video applications were a large complex data is to be operated, it is used in signal processing applications.

Table 1
Difference between single and double precision

SINGLE PRECISION	DOUBLE PRECISION
In single precision, 32 bits are used to represent floating-point number.	In double precision, 64 bits are used to represent floating-point number.
It uses 8 bits for exponent.	It uses 11 bits for exponent.
In single precision, 23 bits are used for mantissa.	In double precision, 52 bits are used for mantissa.
Bias number is 127	Bias number is 1023.
Range of numbers in single precision : 2^(-126) to 2^(+127)	Range of numbers in double precision : 2^(-1022) to 2^(+1023)
This is used where precision matters less.	This is used where precision matters more.
It is used for wide representation.	It is used for minimization of approximation.

III. IEEE 754

IEEE 754 is a standard for floating point established in 1985 then it was updated with major revisions in 2008. It is widely used in software and hardware implementations. The IEEE 754 standard specifies that a single precision number has 1 bit sign, 8 bit exponent and 24 bit of significant precision. The standard defines certain set of rules like:

A. Arithmetic formats: set of binary and decimal floating point data.
B. Interchange formats:encoding that is used to exchange floating point data.
C. Rounding rules: rounding numbers during arithmetic and conversions.
D. Operations: operations like arithmetic and other operations.
E. Exception handling: indication of exceptional handling.

There is also a set of exceptions with a status flag:
1) Invalid: operation of square root of a negative number.
2) Division by zero: returns an infinite result.
3) Overflow: a large number when cannot be represented.
4) Underflow: when operation result is very small or outside the range.
5) Inexact: when a operation returns rounded result by default.

IV. METHODOLOGY

In the proposed floating point architecture consists of the four sub units namely addition, subtraction, multiplication and division. The figure 3 shows the proposed block diagram of floating point unit were 32 bit inputs are taken to perform operation then a multiplexer is used to select operation to be performed and then normalization of the units are done then outputs are taken. These units receive the single precision formatted outputs and performs the operation produces the outputs. The outputs are selected by the selection multiplexer. That multiplexer will be controlled by using the sel signal given by the user. The design is coded in Verilog HDL and tool used to design floating point unit is Xilinx ISE 14.7 and simulator used is ISim. ISim provides a complete, full-featured HDL simulator integrated within ISE.

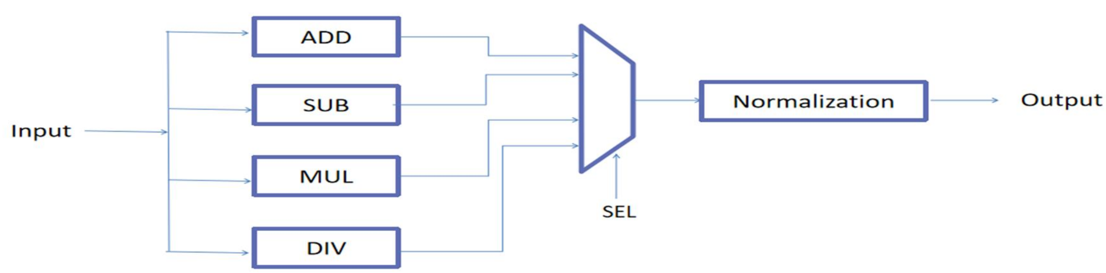


Fig. 3 Block diagram of Floating Point Unit

Table 2
Operation Modes

OP CODE	OPERATION
00	ADDITION
01	SUBTRACTION
10	MULTIPLICATION
11	DIVISION

The table 2 shows the operation modes used in the design methodology while developing this floating point unit, were 00 -means addition, 01 - subtraction, 10 - means multiplication and 11 - means division. The developed floating point unit works efficiently and faster with less time delay.

V. RESULTS AND DISCUSSION

The proposed design has been coded in Verilog and simulation tool used is Xilinx 14.7. A 32 bit number are considered for each result that is addition, subtraction, multiplication and division. Figure 4 shows the result of a 32 bit addition using floating point and the result is obtained. Figure 5 shows the result of a 32 bit multiplication using floating point and the result is obtained. Figure 6 shows the result of a 32 bit division using floating point and the result is obtained. Figure 7 shows the result of a 32 bit subtraction using floating point and the result is obtained. Due to the use of Vedic multiplier the results are improved that is 4.450ns as compared to 5.123ns using an array multiplier [11] that is shown in table 3. A significant improvement of 12% is seen with usage of Vedic multiplier. The results obtained are efficient, less delay and faster computation than the traditional methods like an array multiplier.

Table 3
Result Comparison

Array Multiplier Logic Path Delay (ns)	Vedic Multiplier Logic Path Delay (ns)
5.123	4.450

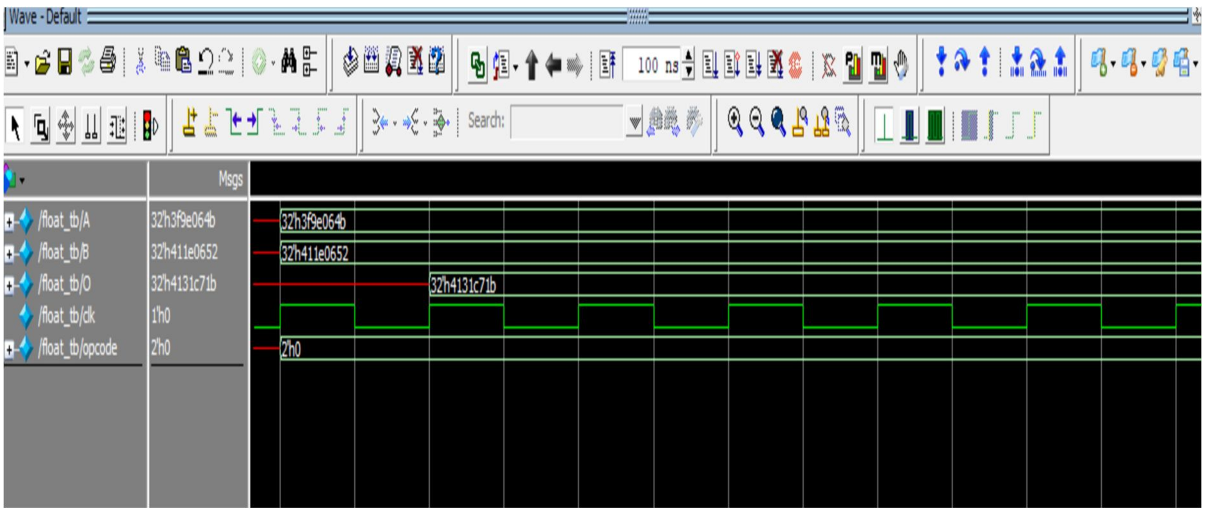


Fig. 4 Result of 32 bit floating point Addition

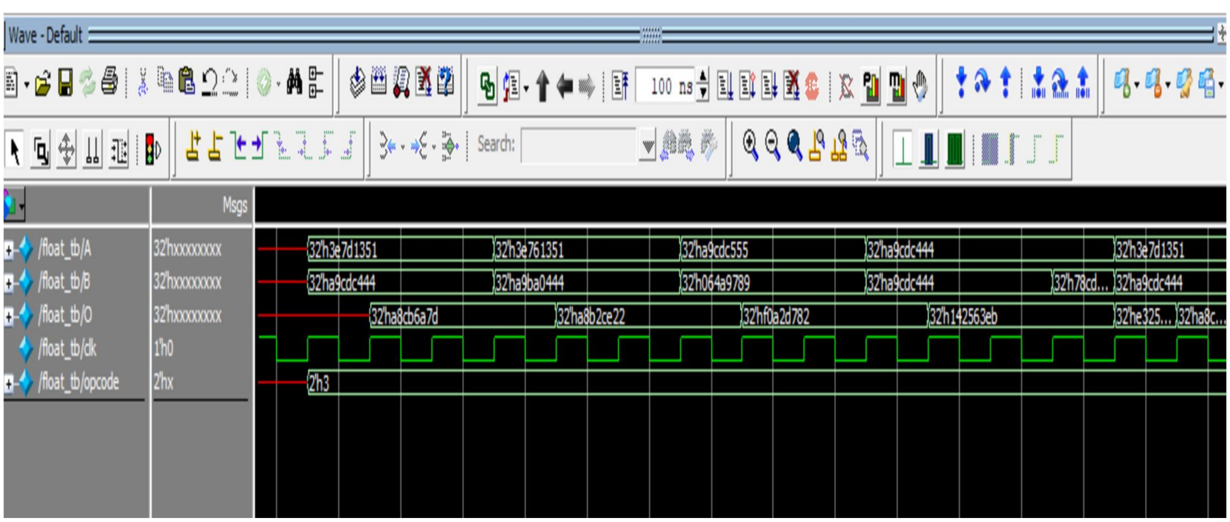


Fig. 5 Result of 32 bit floating point Multiplication

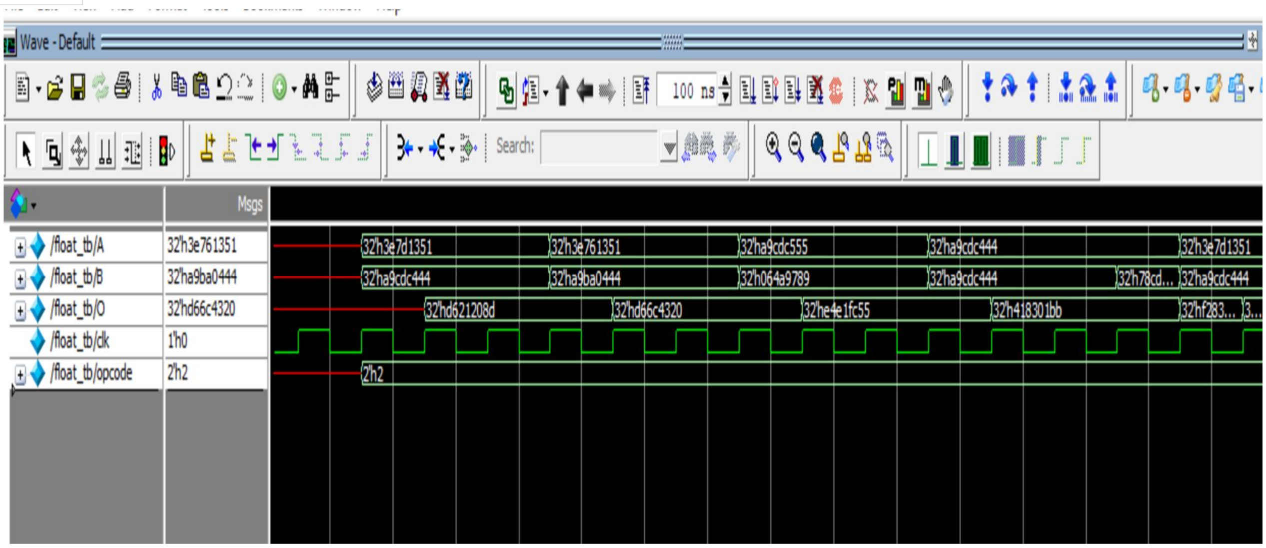


Fig. 6 Result of 32 bit floating point Division

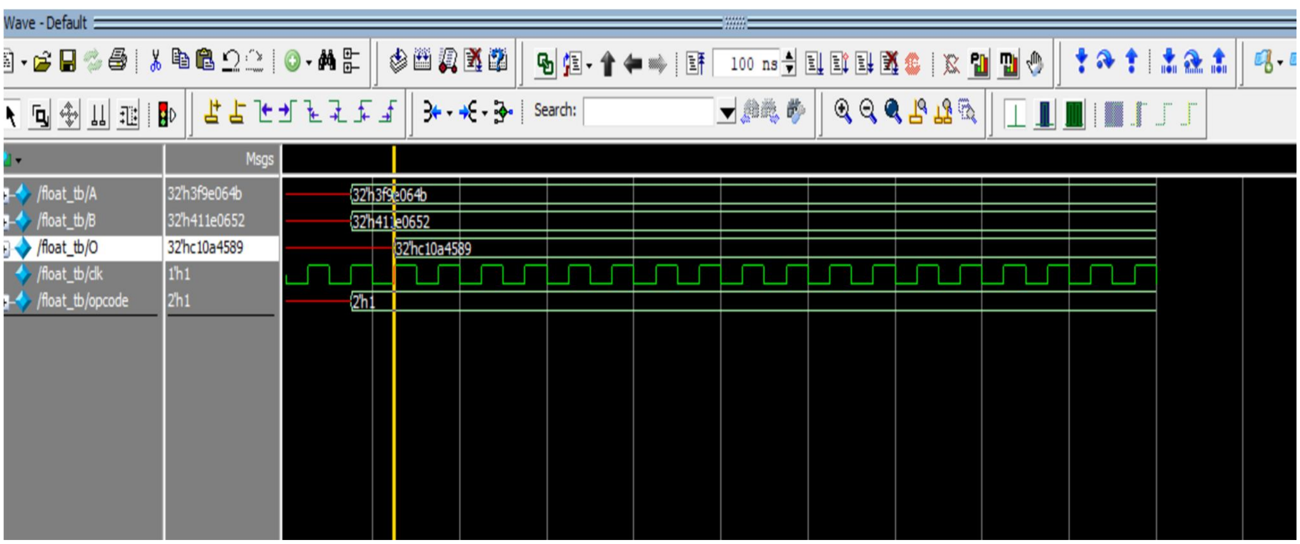


Fig. 7 Result of 32 bit floating point Subtraction

VI. CONCLUSION AND FUTURE SCOPE

An efficient floating point unit block has been designed using Verilog HDL and it is simulated with Xilinx 14.7. In the proposed design efficiency is increased with less computation delay as compared to a traditional method. A 32 bit single precision floating point unit has been designed with IEEE 754 format for addition, multiplication, subtraction and division and the design is verified with simulations using Xilinx ISE tool. Due to the use of Vedic multiplier the results are improved that is 4.450ns as compared to 5.123ns that is shown in table 3. A significant improvement of 12% is seen with usage of Vedic multiplier. The design can be used in mathematical computation, signal processing, graphics and more.

The design can be further optimized in terms of delay, efficiency, fast computation using parallel computation, and efficient truncation and rounding methods to obtain a better floating point unit. Further, a double precision floating point can be used to improve the accuracy and precision. The design can be extended by various other algorithms like faster adders, multipliers and more.

VII. ACKNOWLEDGEMENT

The work was carried out under guidance of Dr. H. V. Ravish Aradhya. I would like to thank my guide, staff, and friends of my institution to have helped me complete the work successfully.

REFERENCES

[1] J. D. Bruguera, "Low Latency Floating-Point Division and Square Root Unit," in IEEE Transactions on Computers, vol. 69, no. 2, pp. 274-287, 1 Feb. 2020, doi: 10.1109/TC.2019.2947899.

[2] J. Sohn and E. E. Swartzlander, "A Fused Floating-Point Four-Term Dot Product Unit," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 63, no. 3, pp. 370-378, March 2016, doi: 10.1109/TCSI.2016.2525042.

[3] J. Kralev, "Design of Floating-Point Arithmetic Unit for FPGA with Simulink®," IEEE EUROCON 2019 -18th International Conference on Smart Technologies, Novi Sad, Serbia, 2019, pp. 1-5, doi: 10.1109/EUROCON.2019.8861860.

[4] T. Krishnan and S. Saravanan, "Design of Low-Area and High Speed Pipelined Single Precision Floating Point Multiplier," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 1259-1264, doi: 10.1109/ICACCS48705.2020.9074366.

[5] Aradhya H.V.R, Aktab M.L.U, Saberi F, “Development of a Random Test Generator for Multi-Core Processor Design Verification,” Proceedings of the lrd International Conference on Electronics and Communication and Aerospace Technology, ICECA-2019, 2019, pp. 1200–1204.

[6] Akella Srinivasa Krishna Vamsi and Ramesh S R, “An Efficient Design of 16 Bit MAC Unit using Vedic Mathematics Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, 2018

[7] Aradhya H.V.R and Goudar S, “Development and Analysis of Parameters to Evaluate Design Partitioning of SoC,” Proceedings of the 2 nd International Conference on Inventive Research in Computing Applications, ICIRCA-2020, 2020, pp. 416–421.

[8] D. L. Prasanna and E. Prabhu, "An Efficient Fused Floating-Point Dot Product Unit Using Vedic Mathematics," 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2019, pp. 12-15, doi: 10.1109/ICOEI.2019.8862718.

[9] L. Kang and C. Wang, "The Design and Implementation of Multi-Precision Floating Point Arithmetic Unit Based on FPGA," 2018 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Xiamen, China, 2018, pp. 587-591, doi: 10.1109/ICITBS.2018.00154.

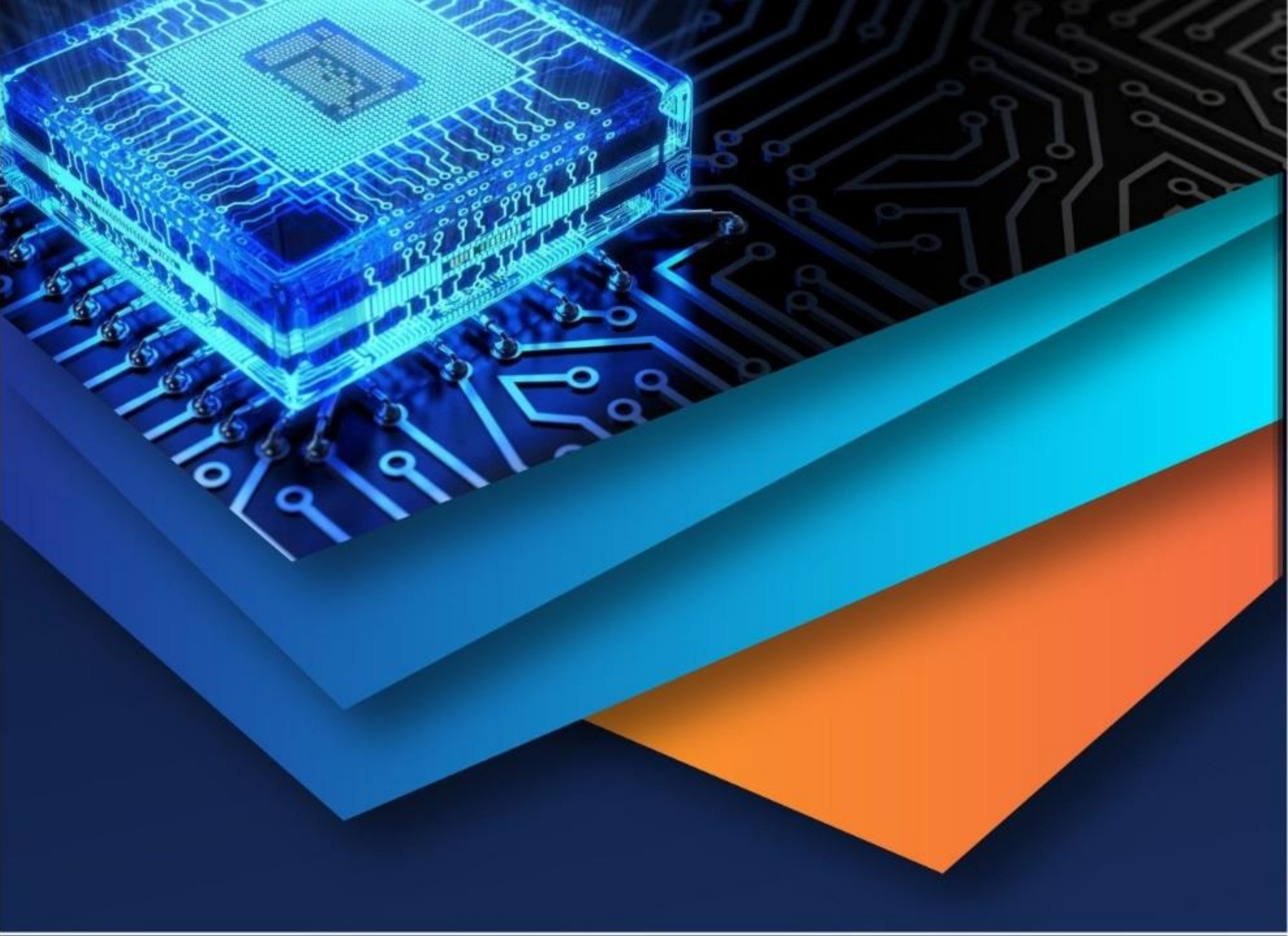
[10] B. Mathis and J. Stine, "A Novel Single/Double Precision Normalized IEEE 754 Floating-Point Adder/Subtracter," 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Miami, FL, USA, 2019, pp. 278-283, doi: 10.1109/ISVLSI.2019.00058.

[11] N. Singh and R. Dhanabal, "Design of Single Precision Floating Point Arithmetic Logic Unit," 2018 4th International Conference on Electrical Energy Systems (ICEES), Chennai, India, 2018, pp. 133-137, doi: 10.1109/ICEES.2018.8442343.

[12] C. R. S. Hanuman and J. Kamala, "Hardware Implementation of 24-bit Vedic Multiplier in 32-bit Floating-Point Divider," 2018 4th International Conference on Electrical, Electronics and System Engineering (ICEESE), Kuala Lumpur, Malaysia, 2018, pp. 60-64, doi: 10.1109/ICEESE.2018.8703551.

[13] A. Burud and P. Bhaskar, "Design and Implementation of FPGA Based 32 Bit Floating Point Processor for DSP Application," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-5, doi: 10.1109/ICCUBEA.2018.8697775

[14] Vaishnavi Kumbargeri, and Ravish Aradhya H V, “Design and Implementation of Logarithmic Multiplier Using FinFETs for Low Power Applications,” in the proceedings of Springer-3rd International Conference on Advanced Trends in Computer Science & Information Technology (ICERECT-18), 24-25 Aug 2018, PES College of Engineering, Mandya, Karnataka, pp 895-902. Part of the Lecture Notes in Electrical Engineering book series (LNEE, volume 545)



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY