International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 07, July 2023)

Literature Review on Karatsuba Algorithm for Multiplication

Manish kumar[1], Dr. Bharti Chourasia[2]

[1]M.Tech Scholar, [2]Prof & HOD ECE, Electronics & Communication Engineering, RKDF IST, Bhopal, India

***Abstract*-- Multiplication is one of the most important operations in computer arithmetic. Division, squaring, and computing reciprocal are only a few of the many operations that use multiplication. Additionally, the utilization of digital signal processing applications including correlation, filtering, frequency analysis, and image processing makes multiplication efficiency vital.In order to make multiplication simpler, algorithms were created to increase efficiency and decrease cost. The effectiveness of the Karatsuba algorithm is examined in this paper in terms of the quantity of multiplication and the overall processing time for various bit lengths. As we all know, algorithms comprise a series of steps/instructions; devised to solve computational problem.VHDL software is used for simulation of implemented proposed algorithm.**

***Keywords*-- Multiplication, Karatsuba algorithm,vhdl, floating Point**

I. Introduction

The Karatsuba algorithm is a recursive algorithm; since it calls smaller instances of itself during execution. According to the algorithm, it calls itself only thrice on n/2-digit numbers in order to achieve the final product of two n-digit numbers. The Karatsuba algorithm is fast multiplication algorithms that uses a divide and conquer approach to multiply two numbers. It was discovered by Anatoly Karatsuba in 1960 and published in 1962. This happens to be the first algorithm to demonstrate that multiplication can be performed at a lower complexity than O(N^2) which is by following the classical multiplication technique. Using this algorithm, multiplication of two n-digit numbers is reduced from O(N^2) to O(N^(log 3) that is O(N^1.585). Multiplication is a very significant arithmetic operation for many signal processing applications which are correlation, convolution, frequency analysis, image processing etc.

II. Literature Review

In this section papers related to karastuba multiplication method are discussed.

X. Fang and L. Li,[1] Algorithms in cryptosystem such as RSA and Diffie-Hellman require the large integer multiplication.

This paper introduces classical Knuth multiplication, Karatsuba multiplication and their time complexity, on the basis of which a new Karatsuba trick is presented and proved to be available in theory and in practice. The experiment result reveals that the improved Karatsuba multiplication is more efficient for implementation of large integer multiplication.

Zoe Siegelnickel Palak Yadav, [2] Algorithms are the foundation of technology today. From medicine to education and beyond, algorithms serve to solve complex problems. This paper explores several types of recursive algorithms and compares them using the conventional notation of time complexity. They analyze algorithms such as the Karatsuba algorithm and the Strassen algorithm, two kinds of algorithms that reduce the time it takes to multiply numbers.

Kowada LAB, Portugal R, Miraglia Herrera de Figueiredo C,[3] Karatsuba discovered the first algorithm that accomplishes multiprecision integer multiplication with complexity below that of the grade-school method. This algorithm is implemented nowadays in computer algebra systems using irreversible logic. In this paper we describe reversible circuits for the Karatsuba's algorithm and analyze their computational complexity. We discuss garbage disposal methods and compare with the well known Bennett's schemes

Luis Antonio Brasil Kowada, Renato Portugal, Celina Miraglia Herrera de Figueiredo, [4] Karatsuba discovered the first algorithm that accomplishes multiprecision integer multiplication with complexity below that of the grade-school method. This algorithm is implemented nowadays in computer algebra systems using irreversible logic. In this paper we describe reversible circuits for the Karatsuba's algorithm and analyze their computational complexity. We discuss garbage disposal methods and compare with the well known Bennett's schemes.

I. V. Vaibhav [5], IEEE drifting point design was a standard arrangement utilized in all handling parts since Binary floating point numbers expansion is one of the principal limits used in cutting edge sign dealing with (DSP) application. In that work VHDLexecution of Floating Point Multiplier utilizing old Vedic science is introduced. The thought for planning themultiplier unit is taken on from antiquated Indian science "Vedas". The Urdhvatriyakbhyam sutra will be utilized for theaugmentation of Mantissa. The sub-current and over stream cases will be dealt with. The contributions to the multiplier in 32 digit design.The multiplier is planned in VHDL or VERILOG and reproduced utilizing Modelsim.

Kowada LAB, Portugal R, Miraglia Herrera de Figueiredo C [6],computerized Signal handling turned into an application to make rapid information handling frameworks like 3direction delivering, 4Generation portable web, and so forth, they really want best processors with elite execution information wayunits and there is a developing requirement for research on elective techniques for signal handling equipment execution. In most frameworks utilizing computerized signal handling Multiply-Accumulate is one of the fundamental capacities. The execution of the entire framework relies upon the exhibition of the MAC units setup.

D. Monniaux et al. [7], because of fast development in monetary, business, and Internet-based applications, there is an expanding want to permit PCs to work on both paired and decimal drifting point numbers. Thus, details for decimal drifting point support are being added to the IEEE-754 Standard for Floating-Point Arithmetic. In this paper, we present the plan and execution of a decimal drifting point viper that is consistent with the current draft modification of this norm. The viper upholds procedure on 64-bit (16-digit) decimal drifting point operands. We give union outcomes showing the assessed region and deferral for our plan when it is pipelined todifferent profundities.

SoumyaHavaldar et al. [8], drifting point number can co-occurrently foster a noticeable scope of numbers and an undeniable degree of accuracy. Duplication of drifting point numbers tracked down broad use in more extensive scope of innovative and business computations It is expected to execute quicker multipliers including restricted region and devouring decreased power. This paper proposes a drifting point multiplier which oversees flood, sub-current and adjusting. The proposed and traditional drifting point multipliers dependent on Vedic arithmetic would be coded in Verilog, Synthesized and Simulated utilizing ISE Simulator.

Xilinx Virtex VI FPGA will be utilized for Hardware acknowledgment and Verification. It is proposed to analyze asset use and timing execution of the proposed multiplier with that of existingat this point.

Ragini Parte et al. [9], drifting point number-crunching has a tremendous applications in DSP, computerized PCs, robots because of its capacity to address tiny numbers and enormous numbers just as marked numbers and unsigned numbers. Notwithstanding intricacy engaged with drifting point number juggling, its execution is expanding step by step. Here we examine the impacts of utilizing three distinct kinds of adders while computing the single accuracy and twofold accuracy drifting point increase. We likewise present the duplication of significand bits by disintegration of operands strategy for IEEE 754 norm.

Ross Thompson et al. [10], this paper examines a streamlined twofold accuracy drifting point multiplier that can deal with both denormalized and standardized IEEE 754 drifting point numbers. Conversations of the improvements are given and looked at versus comparative executions; be that as it may, the principle objective is keeping consistent for denormalized IEEE 754 drifting point numbers while as yet keeping up with elite execution activities for standardized numbers.

Purna Ramesh Addanki et al. [11], Drifting Point (FP) expansion, deduction and duplication are generally utilized in huge arrangement of logical and sign handling calculation. A high velocity drifting point twofold accuracy snake/subtractor and multiplier are executed on a Virtex-6 FPGA. What's more, the proposed plans are consistent with IEEE-754 organization and handles over stream, undercurrent, adjusting and different special case conditions. The viper/subtractor and multiplier plans accomplished the working frequencies of 363.76 MHz and 414.714 MHz with a space of 660 and 648 cuts separately.

Shashank Suresh et al. [12], Drifting point square root is an essential activity in signal handling and different HPC applications. Since this is a costly activity in asset and energy utilization, its proficient execution ought to be of need in future multicores that will confront dull silicon issues. This paper presents a minimal expense, low- power utilization plan to work out the square root utilizing the IEEE754 single- accuracy drifting point design. Two forms of the plan are explored with and without clock gating (CG), separately. Assessment includes FPGA and ASIC advances at 40 and 65 nm. Generous execution development and diminished power utilization are acquired when contrasted with a well-known iterative arrangement.

The ASIC configuration exhibits a lot of lower power utilization, which at 40 nm is lower than that at 65 nm by about a triple. At 40 nm, CG for the ASIC acknowledgment is advocated fundamentally for low action rates.

M. K. Jaiswal et al. [13],Drifting/floating Point (FP) math is generally utilized in huge arrangement of logical and sign handling calculation. Snake/subtractor is one of the normal number juggling activity in these calculation. The plan of FP snake/subtractor is somewhat mind boggling than other FP number-crunching activities. This paper has shown a productive execution of snake/subtractor module on a reconfigurable stage, which is both region just as execution ideal. The proposed plan has advanced the individual complex parts of viper module (like powerful shifter, driving one indicator (LOD),need encoder), to accomplish the better generally execution. Examination withthe best detailedwork has been displayed in the paper, which demonstrates the benefits of proposed plan.

M. Al-Ashrafy et al. [14],this paper depicts a productive execution of an IEEE 754 single accuracy drifting point multiplier focused on for Xilinx Virtex-5 FPGA. VHDL is utilized to execute an innovation autonomous pipelined plan. The multiplier execution handles the flood and undercurrent cases. Adjusting isn't executed to give more accuracy when involving the multiplier in an increase and Accumulate unit. With inertness of three clock cycles the plan accomplishes 301 MFLOPs.The multiplier was confirmed against Xilinx drifting point multiplier center.

D. Sangwan et al. [15],PCs were initially worked as quick, solid and exact registering machines. It doesn't make any difference how huge PCs get, one of their fundamental undertakings will be to consistently perform calculation.. In this manner, most genuine qualities should be addressed in a rough way. The calculations are coded in VHDL and approved through broad reproduction. These are organized with the goal that they give the required execution for example speed and door count. This VHDL code is then incorporated by Synopsys apparatus to produce the entryway level net rundown that can be executed on the FPGA utilizing Xilinx FPGA Compiler.

Andr´e Weimerskirch and Christof Paar,[16] In this work they generalize the classical Karatsuba Algorithm (KA) for polynomial multiplication to (i) polynomials of arbitrary degree and (ii) recursive use. They determine exact complexity expressions for the KA and focus on how to use it with the least number of operations. They develop a rule for the optimum order of steps if the KA is used recursively. They show how the usage of dummy coefficients may improve performance.

Finally they provide detailed information on how to use the KA with least cost, and also provide tables that describe the best possible usage of the KA for polynomials up to a degree of 127. Proposed results are especially useful for efficient implementations of cryptographic and coding schemes over fixed-size fields like GF(p m).

Can Eyupoglu,[17] In computer arithmetic, multiplication is one of the most significant operations. Multiplication is used in many operations such as division, squaring and computing reciprocal. In addition, the efficiency of multiplication is crucial due to the use of digital signal processing applications such as correlation, filtering, frequency analysis and image processing. Karatsuba algorithm is one of the algorithms developed for increasing the efficiency and reducing the cost in order to simplify multiplication. In this study, the performance of Karatsuba algorithm is analyzed in terms of the number of multiplication and the total process time for different bit lengths.

Sudhanshu Mishra,[18] Efficiency in multiplication is very important in applications like signal processing, cryptosystems and coding theory. This paper presents the design of a fast multiplier using the Karatsuba algorithm to multiply two numbers using the technique of polynomial multiplication. The Karatsuba algorithm saves coefficient multiplications at the cost of extra additions as compared to the ordinary multiplication method. The Karatsuba algorithm is more efficient for multiplication of large numbers.

G.Sreelakshmi, K.Ramya Prathima B.Harika Devi [19] among the four arithmetic operations multiplication is one of the basic operations. It can be explained as a repeated addition of multiplicand as the value of the multiplier. The finite field multiplication is the basicoperation in all cryptographic applications. It can be performedby using Conventional, Booth, Montgomery and Karatsuba-Ofman’sdivide-and-conquer technique. The Karatsuba-Ofman multiplierreplaces a multiplication by three ones of half-length operandswhich are performed in parallel. Area, power and delay computation of the proposed multipliers are improved. If ‘n’ is four or more, the three multiplications in Karatsuba's basic step involve operands with fewer than n digits.

Paldurai et al. [20], augmentation of drifting point numbers observed broad use in DSP applications including tremendous reach. The basic part in drifting point augmentation is the increase of mantissas which involves 24*24 piece whole number multiplier for single accuracy drifting point numbers.

The speed of the framework canbe upgraded by working on the speed of duplication. In this paper a 24 cycle Vedic multiplier has been proposed involving 3*3 Vedic multiplier as its essential square. The proposed and regular drifting point multipliers dependent on Vedic science are coded in Verilog, Synthesized and mimicked in ISE Simulator. Greatest combinational way postponement and number of cuts needed on FPGA are looked at for proposed and customary multipliers. The outcomes obviously show that proposed technique incredibly affect working on the speed and diminish the region needed on Spartan 6 FPGA.

Irine Padma et al. [21],to address exceptionally enormous or little qualities, huge reach is needed as the number portrayal is no more suitable. These qualities can be addressed utilizing the IEEE 754 standard based drifting point portrayal. Duplicating drifting point numbers is a basic prerequisite for DSP applications including huge unique reach. The paper depicts the execution and plan of IEEE 754 Pipelined Floating Point Multiplier dependent on Vedic Multiplication Technique. The

R. Sai Siva et al. [23],in this paper we portray an effective execution of an IEEE 754 single accuracy drifting point multiplier utilizing vedic math .The motivation behind utilizing vedic math is because of expansion in the quantity of fractional items in typical augmentation process ,with utilizing vedicmath incomplete items can be decreased so the region also power requirements of the drifting point multiplier can bedecreased proficiently.

Priyanka Koneru et al. [24], a quick and energy effective drifting point unit is constantly required in significant applications like computerized signal handling, picture handling, and ongoing information handling and media applications. As circuits get shrivel, the coordinated plan turns into a basic test as far as clock slant and clock dispersion. One alluring option is to utilize powerful offbeat circuits, which effortlessly oblige these planning disparities. In this paper, a solitary accuracy nonconcurrent drifting point multiplier is carried out utilizing VERILOG equipment depiction language.

III. Conclusion

Karatsuba discovered the first algorithm that accomplishes multiprecision integer multiplication with complexity below that of the grade-school method. In this papermore than 25 paper are review. In Today's computerized word speed is the principle worry for better quality applications like DSP application and installed application.

In these application a large portion of the processing time is devoured by multiplier so multiplier unit should be less tedious and more productive alongside speed we need to think about maturing impacts which hampers multipliers peed.By reading this paper we found that karatsuba algorithm is better than other methods

REFERENCES

[1] X. Fang and L. Li, "On Karatsuba Multiplication Algorithm," The First International Symposium on Data, Privacy, and E-Commerce (ISDPE 2007), Chengdu, China, 2007, pp. 274-276, doi: 10.1109/ISDPE.2007.11.

[2] Zoe Siegelnickel Palak Yadav (2006) Reversible Karatsuba's Algorithm. JUCS - Journal of Universal Computer Science 12(5): 499-511. https://doi.org/10.3217/jucs-012-05-0499

[3] J.Von Zur Gathen and J. Shokrollahi, "Fast arithmetic for polynomials over F2in hardware," in Proceedings of the IEEE Information Theory Workshop (ITW '06), pp. 107–111, Punta del Este, Uruguay, March 2006.

[4] Luis Antonio Brasil Kowada, Renato Portugal, Celina Miraglia Herrera de Figueiredo 10.3217/jucs-012-05-0499

[5] I.V.Vaibhav et al., "Multiplication of many-digital numbers by automatic computers," USSR Academy of Sciences, vol. 145, pp. 293–294, 1962.

[6] Kowada LAB, Portugal R, Miraglia Herrera de Figueiredo C (2006) Reversible Karatsuba's Algorithm. JUCS - Journal of Universal Computer Science 12(5): 499-511. https://doi.org/10.3217/jucs-012-05-0499

[7] Andr´e Weimerskirch and Christof Paar, "Generalizations of the Karatsuba Algorithm for Efficient Implementations", International Association for Cryptologic Research2006

[8] SoumyaHavaldar and Can Eyupoglu* Performance Analysis of Karatsuba Multiplication Algorithm for Different Bit Lengths", World Conference on Technology, Innovation and Entrepreneurship Procedia - Social and Behavioral Sciences 195 (2015) 1860 – 1864.

[9] Ragini Parte and Jitendra Jain, "Analysis of Effects of using Exponent Adders in IEEE- 754 Multiplier by VHDL", International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2015 IEEE.

[10] Ross Thompson and James E. Stine, "An IEEE 754 Double-Precision Floating-Point Multiplier for Denormalized and Normalized Floating-Point Numbers", International conference on IEEE 2015.

[11] Purna Ramesh Addanki, Venkata Nagaratna Tilak Alapati and Mallikarjuna Prasad Avana, "An FPGA based High Speed IEEE-754 double precision floating point Adder/Subtractor and Multiplier using Verilog", in International Journal of Advance Science and Technology, vol. 52, March 2013.

[12] Shashank Suresh, Spiridon F. Beldianu and Sotirios G. Ziavras "FPGA and ASIC square root designs for high performance and power efficiency", in 24th IEEE International conference on Application specific-systems, architecture and processors, June 2013.

[13] M. K. Jaiswal and R. C. C. Cheung, "High Performance FPGA Implementation of Double Precision Floating Point Adder/Subtractor", in International Journal of Hybrid Information Technology, Vol. 4, No. 4, October 2011.

[14] Sudhanshu mishra, “implementation of karatsuba algorithm using polynomial multiplication”, Indian Journal of Computer Science and Engineering (IJCSE) ISSN: 0976-5166 Vol. 3 No. 1 Feb -Mar 2012.

[15] Al-Ashrafy, Mohamed, Ashraf Salem, and Wagdy Anis, “An efficient implementation of floating-point multiplier,” Saudi International Electronics, Communications and Photonics Conference (SIECPC), 24-26 April, pp.1-5, New York: IEEE, (2011) DOI: 10.1109/SIECPC.2011.5876905 Comparative Review of Floating-Point Multiplier 44 Marcus Lloyde George, Geetam Singh Tomar

[16] G.Sreelakshmi, K.Ramya Prathima B.Harika Devi, “AN EFFICIENT VLSI ARCHITECTURE FOR RECURSIVE KARATSUBA-OFMAN MULTIPLIER “,International Journal of Creative Research Thoughts (IJCRT) Volume 6, Issue 2 April 2018 | ISSN: 2320-2882

[17] Kodali, Ravi Kishore, Lakshmi Boppana, and Sai Sourabh Yenamachintala, “FPGA implementation of vedic floating-point multiplier,” IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), 19-21 February, pp.1-4, New York: IEEE, (2015) DOI: 10.1109/SPICES.2015.7091534

[18] Sudhanshu Mishra et al / Indian Journal of Computer Science and Engineering (IJCSE), ISSN : 0976-5166 Vol. 3 No. 1 Feb -Mar 2012

[19] Arish S. and R. K. Sharma, “Run-time reconfigurable multi-precision floating-point multiplier design for high speed, low-power applications,” 2nd International Conference on Signal Processing and Integrated Networks (SPIN), 19-20 February, pp.902-907, New York: IEEE, (2015) DOI: 10.1109/SPIN.2015.7095315

[20] Arish S. and R. K. Sharma, “An efficient binary multiplier design for high speed applications using karatsuba algorithm and urdhva-tiryagbhyam algorithm,” Global Conference on Communication Technologies (GCCT), 23-24 April, pp.192-196, New York: IEEE, (2015) DOI: 10.1109/GCCT.2015.7342650

[21] Paldurai.K and Dr.K.Hariharan "FPGA Implementation of Delay Optimized Single Precision Floating point Multiplier“, 2015 International Conference on Advanced Computing and Communication Systems (ICACCS- 2015), Jan. 05-07-2015, Coimbatore, INDIA.

[22] Irine Padma B.T and Suchitra. K, “Pipelined Floating Point Multiplier Based On Vedic Multiplication Technique,” International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET), ISSN: 2347-6710, Volume-3, Special Issue - 5, July 2014.

[23] R. Sai Siva Teja and A. Madhusudhan,”FPGA Implementation of Low- Area Floating Point Multiplier Using Vedic Mathematics”, International Journal of Emerging Technology and Advanced Engineering (IJETAE), Volume-3, Issue - 12, December 2013, pp.362-366.

[24] Priyanka Koneru, Tinnanti Sreenivasu,and Addanki Purna Ramesh, “Asynchronous Single Precision Floating Point Multiplier Using Verilog HDL,” International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE), ISSN:2278-909X,Volume-2,Issue - 11, November 2014, pp.885-887.

[25] I. V. Vaibhav, K. V. Saicharan, B. Sravanthi and D. Srinivasulu, “VHDL Implementation of Floating Point Multiplier using Vedic Mathematics”, International Conference on Electrical, Electronics and Communications (ICEEC) , ISBN978-93-81693-66-03 , June 2014 pp.110-115.

[26] Ms. Meenu S.Ravi and Mr. Ajit Saraf, “Analysis and study of different multipliers to design floating point MAC units for digital signal processing applications”, International Journal of Research in Advent Technology, (IJRAT), ISSN:2321-9637,Volume-2,Issue-3, March 2014, pp.264-267.

[27] D. Monniaux, “The pitfalls of verifying floating-point computations”, ACM Transaction Programming Language System, Vol. 30, No. 3, pp. 1-12, May 2008.

[28] Soumya Havaldar, K S Gurumurthy, “Design of Vedic IEEE 754 Floating Point Multiplier”, IEEE International Conference on Recent Trends in Electronics Information Communication Technology, May 20-21, 2016, India.

[29] Ragini Parte and Jitendra Jain, “Analysis of Effects of using Exponent Adders in IEEE- 754 Multiplier by VHDL”, International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2015 IEEE. [

[30] Ross Thompson and James E. Stine, “An IEEE 754 Double-Precision Floating-Point Multiplier for Denormalized and Normalized Floating-Point Numbers”, International conference on IEEE 2015.

[31] Even G., S. M. Mueller, and P. M. Seidel, “A dual mode ieee multiplier,” Proceedings of 2nd Annual IEEE International Conference on Innovative Systems in Silicon, 8-10 October, pp.282-289, New York: IEEE, (1997) DOI: 10.1109/ICISS.1997.630271

[32] Sharma, Richa, Manjit Kaur, and Gurmohan Singh, “Design and FPGA implementation of optimized 32-Bit vedic multiplier and square architectures,” International Conference on Industrial Instrumentation and Control (ICIC), 28-30 May, pp.960-964, New York: IEEE, (2015) DOI: 10.1109/IIC.2015.7150883

[33] Anitha P. and P. Ramanathan, “A new hybrid multiplieusing dadda and wallace method,” International Conference on Electronics and Communication Systems (ICECS), 13-14 February, pp.1-4, New York: IEEE, (2014) DOI: 10.1109/ECS.2014.6892623

[34] Sunesh N.V and P Sathishkumar “Design and implementation of fast floating-point multiplier unit,” International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI-SATA), 8-10 January, pp.1-5, New York: IEEE, (2015) DOI: 10.1109/VLSI-SATA.2015.7050478

[35] Cui Xiaoping, Weiqiang Liu, Xin Chen, Earl Swartzlander, and Fabrizio Lombardi, “A modified partial product generator for redundant binary multipliers,” IEEE Transactions on Computers, vol.65, no.4, pp.1165-1171, (2015) DOI: 10.1109/TC.2015.2441711

[36] Huntsman C. and D. Cawthron, “The MC68881 floating-point coprocessor,” IEEE Micro, vol.3, no.6, pp.44- 54, (1983) DOI: 10.1109/MM.1983.291185

[37] Thapliyal, Himanshu, and M. B. Srinavas, “A novel time-area-power efficient single precision floating multiplier,” Proceedings of MAPLD 16-18 June, pp.1-3, New York: IEEE, (2005)

[38] Siddamal, Saroja V., R. M. Banakar, and B. C. Jinaga, DELTA 2008. “Design of high-speed floating-point multiplier,” 4th IEEE International Symposium on Electronic Design, Test and Applications, 23-25 January, pp.285-289, New York: IEEE, (2008) DOI: 10.1109/DELTA.2008.19

[39] Nachtigal, Michael, Himanshu Thapliyal, and Nagarajan Ranganathan, “Design of a reversible single precision floating-point multiplier based on operand decomposition,” 10th IEEE Conference on Nanotechnology (IEEENANO), 17-20 August, pp.233-237, New York: IEEE, (2010) DOI: 10.1109/NANO.2010.5697746

[40] MG (Mentor Graphics), Precision Synthesis User's Manual. Ottawa: MG, (2010)

[41] Mehta, Anand, C. B. Bidhul, Sajeevan Joseph, and P. Jayakrishnan, "Implementation of single precision floating-point multiplier using karatsuba algorithm," International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), 3-5 November, pp.254-256, New York: IEEE, (2013) DOI: 10.1109/ICGCE.2013.6823439

[42] Paldurai K. and K. Hariharan, "FPGA Implementation of Delay Optimized Single Precision Floating-Point Multiplier," International Conference on Advanced Computing and Communication Systems, 5-7 January, pp.1-5, New York: IEEE, (2015) DOI: 10.1109/ICACCS.2015.7324094

[43] Mano M. Morris and Charles R. Kime, Logic and Computer Design Fundamentals. New Jersey: Prentice Hall, (1997)

[44] Gupta, Aman, Satyam Mandavalli, Vincent J. Mooney, Keck-Voon Ling, Arindam Basu, Henry Johan, and Budianto Tandianus, "Low power probabilistic floating-point multiplier design," 2011 IEEE Computer Society Annual Symposium on VLSI, 4-6 July, pp.182-187, New York: IEEE, (2011) DOI: 10.1109/ISVLSI.2011.54

[45] Beohar, Salty, and Sandip Nemade, "VHDL implementation of self-timed 32-bit floating-point multiplier with carry look ahead adder," International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), 25-27 May, pp.772-775, New York: IEEE, (2016) DOI: 10.1109/ICACCCT.2016.7831743

[46] Cheng, Fu-Chiung, Stephen H. Unger, Michael Theobald, and Wen-Chung Cho, "Delay-insensitive carry-look ahead adders," Proceedings of 10th International Proceedings VLSI Design, Conference, 4-7 January, pp.37- 63, New York: IEEE, (1997)

[47] Jaiswal, Manish Kumar, C. Ray, and C. Cheung, "Area-efficient FPGA implementation of quadruple precision floating-point multiplier," IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 21-25 May, pp.376-382, New York: IEEE, (2012) DOI: 10.1109/IPDPSW.2012.46

[48] Ramesh, Addanki Purna, A. V. N. Tilak, and A. M. Prasad, "An FPGA-based high speed IEEE-754 double precision floating-point multiplier using verilog," International Conference on Emerging Trends in VLSI, Embedded System, Nano Electronics and Telecommunication System (ICEVENT), 7-9 January, pp.1-5, New York: IEEE, (2013) DOI: 10.1109/ICEVENT.2013.6496575

[49] Rao, Y. Srinivasa, M. Kamaraju, and D. V. S. Ramanjaneyulu, "An FPGA implementation of high speed and area efficient double-precision floating-point multiplier using urdhva tiryagbhyam technique," Conference on Power, Control, Communication and Computational Technologies for Sustainable Growth (PCCCTSG), 11-12 December, pp.271- 276, New York: IEEE, (2015) DOI: 10.1109/PCCCTSG.2015.7503923

[50] Shanmugapriyan S. and K. Sivanandam, "Area efficient run time reconfigurable architecture for double precision multiplier," IEEE 9th International Conference on Intelligent Systems and Control (ISCO), 9-10 January,pp.1-6, New York: IEEE, (2015) DOI: 10.1109/ISCO.2015.7282355

[51] Lei Kang and Yan Xiao-Ying, "Design and implementation for quadruple precision floating-point multiplier based on fpga with lower resource occupancy," 5th International Conference on Intelligent Systems Design and Engineering Applications (ISDEA), 15-16 June, pp.326-329, New York: IEEE, (2014) DOI: 10.1109/ISDEA.2014.80