International Journal of Scientific Research in Science and Technology
Available online at : www.ijsrst.com
Print ISSN: 2395-6011 | Online ISSN: 2395-602X
doi : https://doi.org/10.32628/IJSRST

Design And Implementation of an M-Term Karatsuba-Like Polynomial Multiplier for Finite Field Arithmetic

P. Suresh babu[1], R. Susmitha[2], C. Sasidhar[3], B. Bhavana Rameswari[4], E. Rukesh[5], C. Lehitha[6]

[1]Assistant Professor, Department of Electronics and Communication Engineering, SV College of Engineering (SVCE), Tirupati, A.P. India
[2,3,4,5,6]UG Students, Department of Electronics and Communication Engineering, SV College of Engineering (SVCE), Tirupati, A.P. India

ARTICLEINFO

Article History:

Accepted: 03 March 2024
Published: 15 March 2024

Publication Issue :
Volume 11, Issue 2
March-April-2024
Page Number :
210-216

ABSTRACT

The abstract of a Karatsuba multiplier employing a combination of 32-term Karatsuba algorithm, Schoolbook multiplication, adders, shifting bits, and recursive Karatsuba operations would highlight the hybrid nature of the approach and its potential advantages in terms of efficiency and speed. The Karatsuba algorithm is renowned for its divide-and-conquer strategy, which efficiently breaks down large multiplication tasks into smaller sub-problems, thereby reducing computational complexity. The utilization of a 32-term Karatsuba variant enhances its ability to handle larger operands effectively. Schoolbook multiplication, a conventional multiplication technique, is incorporated into the algorithm to leverage its simplicity and straightforward implementation, particularly for smaller operand sizes. Adders, fundamental arithmetic units, play a pivotal role in the multiplier, facilitating the addition of partial products and intermediate results efficiently. Shifting bits are utilized to manipulate binary numbers, aiding in the alignment of operands and computation of partial products. The integration of these techniques creates a hybrid multiplier architecture that capitalizes on the strengths of both Karatsuba and Schoolbook multiplication methods, resulting in improved performance and efficiency. This approach holds promise for accelerating multiplication operations in various computational tasks, including digital signal processing, cryptography, and computer arithmetic.

Keywords : Karatsuba multiplier, Schoolbook multiplication (SBM), Adder, Shifting bits, Partial products

I. INTRODUCTION

Multiplication is a fundamental arithmetic operation extensively used in various computational tasks, ranging from digital signal processing to cryptography. Traditional multiplication algorithms, such as Schoolbook multiplication, exhibit linear time complexity, which becomes a bottleneck when dealing with large operands. To overcome this

limitation, more efficient algorithms have been developed, among which the Karatsuba multiplier stands out.

The Karatsuba algorithm, introduced by Anatolii Alexeevitch Karatsuba in 1960, employs a divide-and-conquer strategy to reduce the number of required operations for multiplication, leading to improved efficiency. By breaking down large multiplication tasks into smaller, more manageable sub-problems, Karatsuba multiplication significantly reduces the computational complexity compared to traditional methods. This introduction presents a novel approach to Karatsuba multiplication by leveraging a 32-term Karatsuba variant, Schoolbook multiplication, adders, and shifting bits. This hybrid architecture aims to capitalize on the strengths of both Karatsuba and Schoolbook multiplication techniques, enhancing computational efficiency and speed. The inclusion of a 32-term Karatsuba variant enhances the algorithm's ability to handle larger operands efficiently. Schoolbook multiplication, a conventional technique, is utilized alongside Karatsuba to handle smaller operand sizes effectively. Adders, fundamental arithmetic units, are incorporated into the architecture to facilitate the addition of partial products and intermediate results, crucial for the multiplication process.Shifting bits play a vital role in manipulating binary numbers, aiding in operand alignment and partial product computation, thereby optimizing the multiplication process.By combining these techniques, the proposed Karatsuba multiplier architecture offers a versatile and efficient solution for high-speed multiplication tasks, making it suitable for various applications in computer arithmetic, cryptography, and digital signal processing.

With the ever-growing expansion of modern information technologies in almost every field, the number of threats and the importance of information security are increasing day by day.

Cryptography systems play a crucial role in ensuring the safety and security of information.In these systems, a fundamental and frequently used operation that determines the overall speed and cost of systems is field multiplication.

The organizational framework of this study divides the research work in different sections. The Literature survey is presented in section 2. In section 3 and 4 discussed about Existing and proposed system methodologies. Further, in section 5 shown Simulation Results is discussed and Conclusion and future work are presented by last sections 6.

II. LITERATURE SURVEY

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

X. Fang and L. Li,[1] Algorithms in cryptosystems 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.

J.Von Zur Gathen and J. Shokrollahi,[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.

M. K. Jaiswal and R. C. C. Cheung [4] Karatsuba discovered the first algorithm that accomplishes multiprecision integer multiplication with complexity below that of the gradeschool 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.

Shashank Suresh [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 the multiplier unit is taken on from antiquated Indian science "Vedas". The Urdhvatriyakbhyam sutra will be utilized for the augmentation 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.

Purna Ramesh [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.

Ross Thompson [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 IEEE754 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.

Ragini Parte and Jitendra Jain [8], drifting point number can cooccurrently 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.

SoumyaHavaldar and Can Eyupoglu. [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.

Andr´e Weimerskirch and Christof Paar [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.

III. EXISTING METHOD

In the existing method, it uses Karatsuba Multiplication. The Karatsuba algorithm is a fast multiplication algorithm that was discovered by Anatolii Alexeevitch Karatsuba in 1960. The algorithm is used to multiply two n-digit numbers using a recursive approach.

Karatsuba algorithm

Step- 1. Input:

- Two n-digit numbers, let's call them x and y.

Step-2. Base Case:

- If n is small (e.g., a threshold value), switch to a simple multiplication algorithm (e.g., grade school multiplication) to handle the multiplication.

Step-3. Divide:

- Divide the input numbers, x and y, into two halves. This creates four smaller sub-problems.

Step-4. Recursive Multiplication:

- Recursively compute three products.

Step-5. Combine:

- Combine the results of the recursive multiplications to obtain the final product.

Step-6. Efficiency:

- The Karatsuba algorithm reduces the number of recursive calls compared to traditional multiplication, making it more efficient for large numbers.

Step-7. Complexity:

- The time complexity of the Karatsuba algorithm is better than the traditional multiplication algorithm. It has a time complexity of approximately O(n^log2(3)), which is an improvement over the O(n^2) time complexity of traditional multiplication.

Step-8. Optimizations:

- To further enhance performance, optimizations like using FFT (Fast Fourier Transform) or Toom-Cook multiplication may be applied for even larger numbers.

While Karatsuba multiplication is a powerful algorithm that reduces the number of multiplications in polynomial multiplication, it is not without drawbacks. Here are some drawbacks of the existing method of Karatsuba: Overhead for Small Input Sizes, Increased Constant Factors, Memory Requirements, Complexity of Implementation, and Threshold Sensitivity.

The block diagram of existing method is shown in Figure 1.

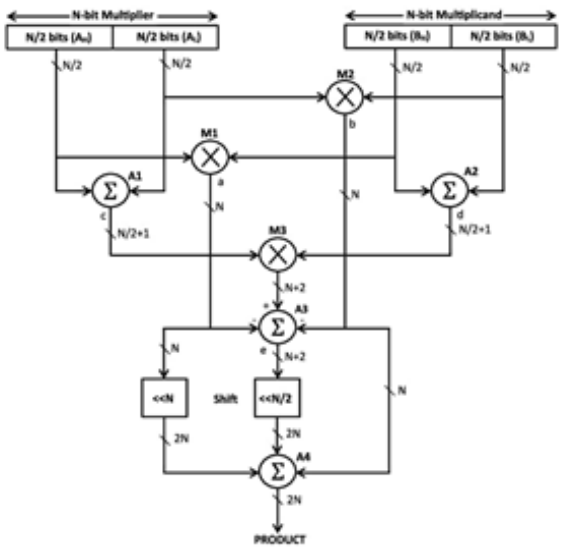

Figure 1. Block diagram of existing method

The Karatsuba algorithm is a fast multiplication algorithm that multiplies two numbers using a recursive approach. The basic idea behind the Karatsuba algorithm is to break down the multiplication of large numbers into smaller multiplications, reducing the number of required multiplications and additions. The Karatsuba

algorithm can also be applied for polynomial multiplication, which is especially useful in polynomial arithmetic, such as in error-correcting codes or cryptograph.

IV. PROPOSED METHOD

An M-term Karatsuba-like multiplier breaks the operands into smaller size operands and uses a number of submultipliers to recursively calculate the product. M term Karatsuba uses a similar concept as KOM but splits to a higher number of equal parts. For the rest of this article, we assume that each operand is split into M number of polynomials with equal length. The block diagram is shown in figure 2.

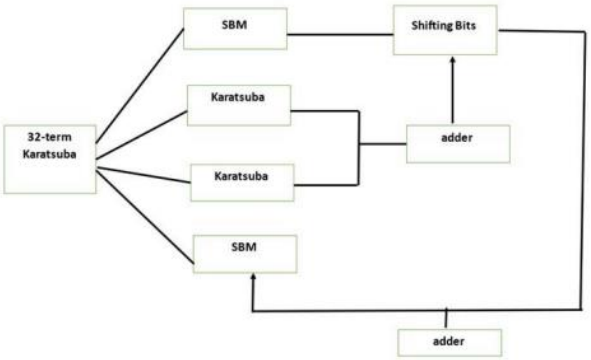

Figure 2. Block diagram of Karatsuba multiplier

SBM:

Schoolbook multiplication (SBM) is the most straightforward method for polynomial multiplication, but it can be computationally expensive for large polynomials.

Karatsuba:

Karatsuba is a divide-and-conquer algorithm that reduces the number of recursive multiplications by expressing the product of two polynomials as three multiplications instead of four.

Adder:

An adder circuit is central to cryptography circuits that perform addition.

Shifting Bit:

Shift registers are at the heart of cryptography and error correction. In cryptography, they are the main tool for generating long pseudorandom binary sequences which can be used as keys for two communicating parties in symmetric cryptography.

Efficient Multiplier:

Compared to the schoolbook, M-term Karatsuba has the disadvantage of higher time complexity due to the larger number of recursive products. However, it is considerably more efficient because of the lower area–delay product (ADP). Efficient multiplication algorithms are crucial in various applications, including cryptography, signal processing, and numerical computing. The choice of a multiplication algorithm depends on factors such as the size of the operands, the specific characteristics of the data, and the computational resources available. Karatsuba reduces the number of multiplications by expressing the product as three multiplications instead of four. Efficient for moderately sized operands. The block diagram of efficient karatsuba multiplier is shown in figure 3.

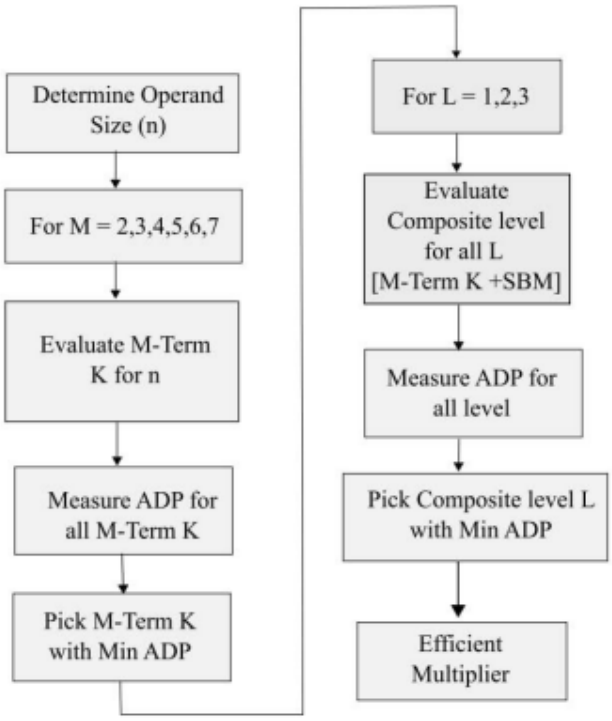

Figure 3. Block diagram of efficient karatsuba multiplier

Compared to the school book, M-term Karatsuba has a disadvantage of higher time complexity due to the larger number of the recursive product. However, it is considerably more efficient because of the lower area–delay product (ADP). A methodology to achieve an efficient composite finite field multiplier is given in Fig. 3. This method utilizes the M-term Karatsuba-like multiplier at the upper bound and SBM at the lower bound of recurrent stages. It reduces the higher time complexity of the M-term Karatsuba-like multiplier and optimizes higher combinational delay.

IMPLEMENTATION

The implementation of this methodology was handled in two phases. 1) Phase A: The M-term Karatsuba-like multiplier for operand size of "n" was implemented for M = 2, 3, 4, 5, 6, and 7. Based on the ADP, an efficient M-term multiplier was selected. 2) Phase B: A composite multiplier for the operand size of "n" was constructed for different levels of 1, 2, and 3. The composite M-term multiplier with a minimum ADP was adopted as the most efficient multiplier.

V. RESULTS AND DISCUSSIONS

RTL Schematic

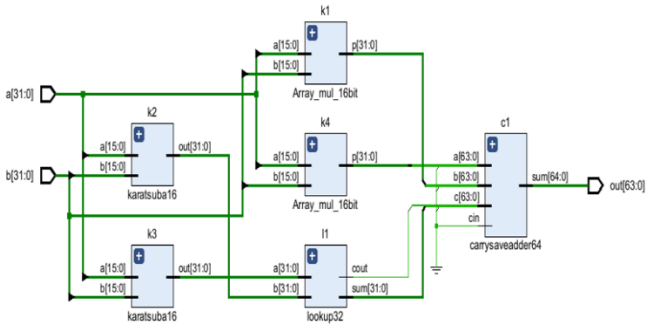

Figure 4. RTL Schematic

The implementation was simulated in Xilinx Vivado 2018.3. The results are shown in below figure 5, Area and delay of different paths of Karatsuba 32 are shown in Table 1 and Table 2 respectively.

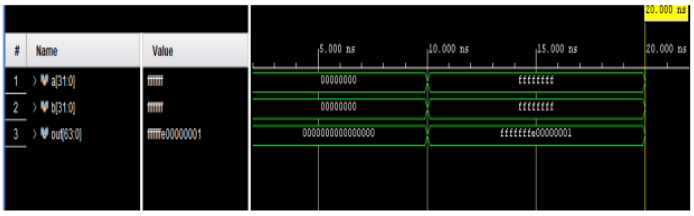

Figure 5. Result obtained after simulation

Table 1. Area of Karatsuba 32

Hierarchy

Name	Slice LUTs (134600)	Bonded IOB (400)
karatsuba32	2077	128

Table 2. Delay values for different Paths

Unconstrained Paths - NONE - NONE - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Cl
Path 1	∞	37	38	94	a[3]	out[61]	26.248	6.917	19.331	∞	input port
Path 2	∞	37	38	94	a[3]	out[62]	26.248	6.917	19.331	∞	input port
Path 3	∞	37	38	94	a[3]	out[63]	26.248	6.917	19.331	∞	input port
Path 4	∞	37	38	94	a[3]	out[60]	26.230	6.917	19.313	∞	input port
Path 5	∞	36	37	94	a[3]	out[57]	25.830	6.812	19.018	∞	input port
Path 6	∞	36	37	94	a[3]	out[58]	25.778	6.812	18.966	∞	input port
Path 7	∞	36	37	94	a[3]	out[59]	25.778	6.812	18.966	∞	input port
Path 8	∞	35	36	94	a[3]	out[53]	25.381	6.710	18.671	∞	input port
Path 9	∞	35	36	94	a[3]	out[55]	25.381	6.710	18.671	∞	input port

VI. CONCLUSION AND FUTURE SCOPE

CONCLUSION

In this work, first M-term Karatsuba-like binary multipliers were analysed in terms of space and time complexities for different values of M and various operand sizes (n). Performance parameter's trends were pictured for the Xilinx Artix FPGA device. Later, a novel composite method is introduced to take advantage of the low space complexity of M-term Karatsuba-like and low time complexity SBM. The proposed method was extensively tested on different FPGAs to attain the improvement graph over other similar works. In FPGA devices, implementation results show that the composite method requires 11% additional resources and drops the delay complexity 26% lower, and it is 15% more efficient in ADP than standard KOM. This work achieved the suitable trade-off between space and time complexities, which minimizes the ADP requirement of the multiplier.

FUTURE SCOPE

The future scope of an m-term Karatsuba multiplier lies in its potential applications in high-performance computing, particularly in areas where the multiplication of large numbers is a bottleneck. Here are some potential areas where m-term Karatsuba multipliers could find significance:

Cryptography: Cryptographic algorithms such as RSA, ECC (Elliptic Curve Cryptography), and various hashing algorithms rely heavily on large integer multiplication. Efficient multiplication techniques like Karatsuba can significantly improve the performance of these cryptographic algorithms.

ACKNOWLEDGEMENT

It gives us great pleasure to present the preliminary project report. I would like to take this opportunity to thank my guide Mr.P.Suresh Babu M.Tech,(Ph.D) Assistant Professor, and Dr. D. Srinivasulu Reddy, Ph.D., Professor, & Head of the Department (HOD) of Electronics and Communication Engineering, SV College of Engineering (SVCE) (Autonomous), Tirupati, Andhra Pradesh India, for giving me all the help and guidance I needed. I am really grateful for their kind support and valuable suggestions. Thank you all!

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 Siegel nickel 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. 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.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. Andre Weimerskirch and Christof Paar, “Generalizations of the Karatsuba Algorithm for Efficient Implementations”, International Association for Cryptologic Research2006.

Cite this article as :

T Chandra Sekhar Rao, S Divya, M Vennela, S Kavya, S Anil Yadav, R Yashoda, "Design And Implementation of an M-Term Karatsuba-Like Polynomial Multiplier for Finite Field Arithmetic ", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 11 Issue 2, pp. 210-216, March-April 2024.

Journal URL : https://ijsrst.com/IJSRST524112235