iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IV **Month of publication:** April 2024

DOI: https://doi.org/10.22214/ijraset.2024.59866

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

High Performance M-Term Karatsuba-Like Multiplier for Finite Field Arithmetic

M. Devendra Sai[1], M. Swathi[2], M. Priyanka[3], Mrs. S. Karuna[4]
[1, 2, 3] *Student,* [4]*Assistant Professor, Department of Electronics and Communication Engineering, Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru, Krishna Dt., Andhra Pradesh*

Abstract: ***Finite field multiplication plays a crucial role in cryptographic circuits due to its widespread application. However, building circuits for these multiplications poses significant challenges due to their complexity. To mitigate this, the Karatsuba algorithm is employed, dividing each number into n/2 bits to reduce space complexity. While this approach reduces space complexity, it also increases time complexity. In our research, we introduce a hybrid approach, implementing a Karatsuba-like multiplier that combines elements of both Karatsuba and SBM (school book multiplication) techniques. Here in the proposed design, we replace the Array multiplier with a Wallace tree multiplier to further enhance design performance. This combination effectively reduces both time and space complexity. Our findings, based on reported device utilization and latency, demonstrate that the proposed multiplier outperforms the standard Karatsuba multiplier in terms of speed and efficiency, particularly in the area–delay product metric.***

Keywords: Binary polynomial multiplier, field-programmable gate array (FPGA), finite field multiplication, Wallace tree multiplier, M-term Karatsuba-like.

I. INTRODUCTION

The continuous proliferation of modern information technologies across various sectors has led to a rise in the number of threats and the significance of information security. Cryptography systems are instrumental in ensuring the protection and security of information. Within these systems, finite field multiplication stands out as a crucial and frequently used operation that influences the overall speed and cost of systems. Therefore, the efficiency of the multiplier holds paramount importance. Among the range of polynomial multiplication algorithms, the school-book multiplication (SBM) emerges as the most elementary form of multiplication. The complexity of SBM for two polynomials of n-1 degree is O(n2).

To enhance the efficiency of multiplication, a different type of algorithms has been put forth by researchers. One well-known algorithm is the Karatsuba-Ofman multiplier (KOM), which presents a recursive multiplicative approach with lower space complexity [O (nlog2 3)] compared to the traditional SBM. This strategy involves dividing the operands into lower and upper parts and utilizing three submultipliers to compute the product. Despite the resource efficiency achieved by KOM, it suffers from increased delay when compared to SBM due to its submultiplier-based recursive framework. To tackle this challenge, several Karatsuba modifications and diverse implementation approaches have been proposed.

An innovative solution called Overlap-free Karatsuba was introduced to eliminate the higher combinational delay of the general Karatsuba method. A low-complexity Karatsuba multiplier implemented a fresh approach to reduce the complex register configuration in current systolic implementation, thereby decreasing area and power consumption. Meanwhile, Samanta et al. devised a modified Karatsuba implementation tailored for 8-bit operands, where terms are segregated into different formats to minimize operational latency. Additionally, Li et al. proposed a novel non-recursive Mastrovito multiplier for GF(2m) using an n-term Karatsuba algorithm (KA). Furthermore, Chiou-Yng et al. demonstrated an effective digit-level parallel-in–serial-out (PISO) multiplier with sub quadratic space complexity through the utilization of the overlap-free Karatsuba multiplication algorithm.

The introduction of M-term Karatsuba-like algorithms has garnered significant attention in recent times. These algorithms allow for the division of operands into a greater number of terms compared to the standard two-term Karatsuba approach, consequently reducing the recurrence stages and enhancing multiplication speed. Their optimization of the existing M-term Karatsuba-like algorithm yielded reduced size and depth compared to prior works.

The main contributions of the outlined study are as follows:

1) Conducting gate-level space and time complexity analyses for extended term-based Karatsuba-like algorithms.
2) Devising a strategic plan to develop an effective finite field multiplier using the M-term Karatsuba-like approach.
3) Experimentally evaluating the proposed hardware on FPGA with reduction in delay and area compared to KOM.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

II. AIM AND OBJECTIVES

A. *Aim*

To design a 32-bit Binary Polynomial Multiplier.

B. *Objectives*

1) To reduce the delay of the multiplier.
2) To reduce the area of the multiplier.

III. BACKGROUND

In the realm of finite field arithmetic, the process of multiplication involves utilizing a binary polynomial multiplier, followed by a modular reduction with an irreducible polynomial. The complexities of space and delay in implementing this multiplier are determined by its multiplicative and additive costs. The assessment of space complexity involves the number of combinational gates needed for implementation, while delay complexity is gauged by the linear sum of standard gate delays. This section evaluates the space and delay complexities of the SBM method and the M-term Karatsuba-like approach. By utilizing the both multiplication techniques SBM and Karatsuba we are implementing a 32-bit Binary Polynomial Multiplier.

A. *SBM Algorithm*

In School Book Multiplication, let us consider A and B as two-degree binomial

$$A(x) = A_1x + A_0$$
$$B(x) = B_1x + B_0$$

By multiplying it, we get

$$A(x).\ B(x) = A_1.B_1x^2 + (A_1.B_0 + A_0.B_1)x + A_0.B_0$$

This is the normal multiplication, which are constructed using AND and XOR gates. In this SBM, there are different types of multipliers we are using in daily life, some of them are Array Multiplier, Vedic Multiplier, Wallace Multiplier, Dadda Multiplier etc., for the multiplication. By using this SBM algorithm, we can reduce the time complexity of the multiplier, but required area is more in this SBM.

B. *M-term Karatsuba-Like Multipliers*

The M-term Karatsuba-Like multipliers are used to reduce the area complexity of the multiplier. By the name itself it is known as divide and conquer algorithm. For this multiplier we use Karatsuba algorithm, which is used to reduce the maximum amount of complexity by dividing each number into n/2 bits. Let us consider ab and cd we two numbers, by multiplying we get partial products as ac, ad, bc, bd. For MSB and LSB partial products we undergo into SBM and remaining two partial products will again undergo into Karatsuba algorithm until it gets the lowest power of bits. By this algorithm we can reduce the area, but time complexity will get increased.

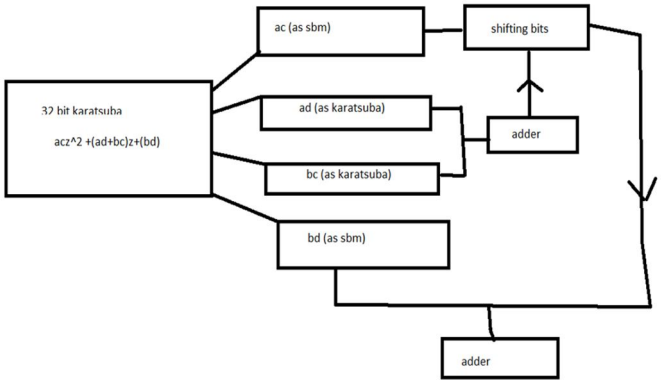


Fig 1. Karatsuba Algorithm

This Karatsuba algorithm will get divided until it gets the lowest power of bits. After reaching the lowest power, it performs the Karatsuba multiplier operation to that number.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IV Apr 2024- Available at www.ijraset.com

IV. EXISTING SYSTEM

There were few existing systems like complete SBM (school book multiplication) and complete Karatsuba multiplication and few Vedic multiplication techniques used in many cryptographic circuits. Existing method is named as Karatsuba like multiplier. In this existing system we take both Karatsuba and Array Multiplier (as SBM). By using these systems, a 32-bit binary polynomial multiplier is constructed, which reduces the time complexity and area complexity. As increasing the day-by-day technology, we need to decrease the time and area of the required multiplier.

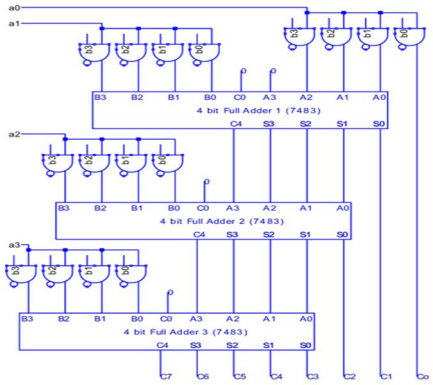


Fig 2. Array Multiplier

By using this Array multiplier (as SBM), it takes some time for generating partial products and adding them. So to overcome the situations we had proposed another multiplier which is used to reduce the area and delay of the required multiplier than in the existing system.

A. Disadvantages

1) Considering Array multiplier as SBM, makes design less efficient.
2) Consumes more area
3) More delay

V. PROPOSED METHOD

To overcome the disadvantages, we had proposed Wallace tree multiplier as SBM. A Wallace tree multiplier is an improved version of tree-based multiplier architecture to reduce the latency. Wallace tree multiplier is useful in all respect like delay, speed, complexity, area, power consumption. The Wallace scheme is one of the parallel multiplier schemes that essentially minimize the number of adder stages required to perform the summation of partial products. This is achieved by using full and half adder to reduce the number of rows at each summation stage, this Wallace multiplication has regular and less complex structure.

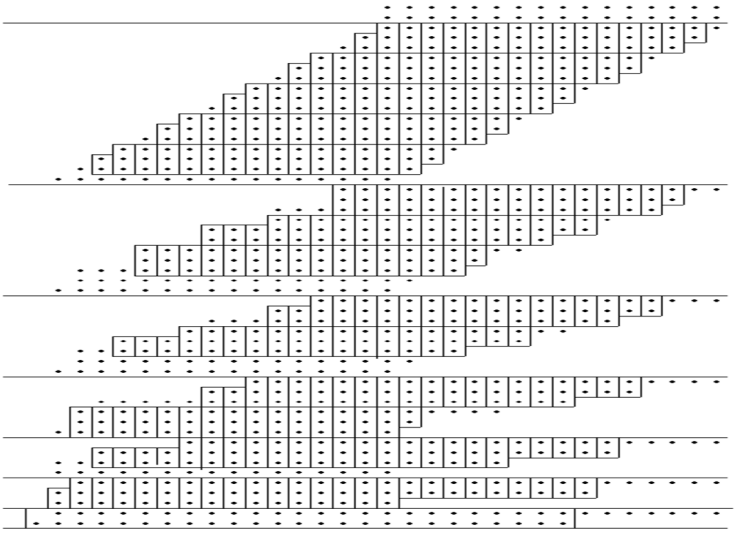


Fig 3. 16×16 Wallace Tree Multiplier

VI. METHODOLOGY

Wallace multiplier will overcome the drawbacks of the Array multiplier. In this multiplier, we get total 16 partial products. Then we reduce these partial products by applying full adders at every 3 stages. By performing this process continuously, we get final product value. This Wallace operation is performed for MSB and LSB as mentioned in the Karatsuba algorithm, remaining bits are performed by Karatsuba multiplier. At finally we add all these values by shifting bits according to their bit preferences. Here, Wallace multiplier will have time complexity and Karatsuba multiplier will have space complexity. By using this both Wallace tree multiplier (as SBM) and Karatsuba-like multiplier we had constructed a 32-bit Binary Polynomial Multiplier.

A. *Advantages*

1) Drastically reduces time and space complexity compared to existing Karatsuba-like multipliers.
2) Decreases complexity in circuit implementation.

VII. RESULTS

A. *RTL Schematic*

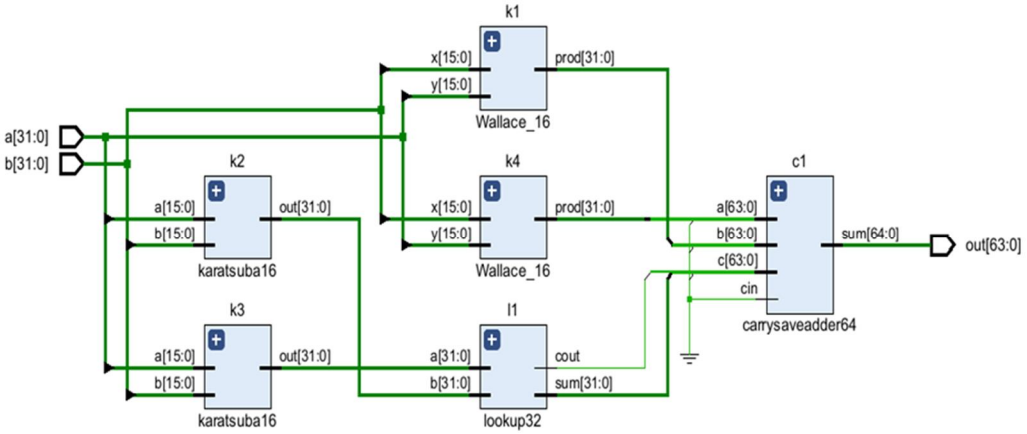


Fig 4. RTL Schematic of 32-bit Binary Polynomial Multiplier

B. *Technology Schematic*

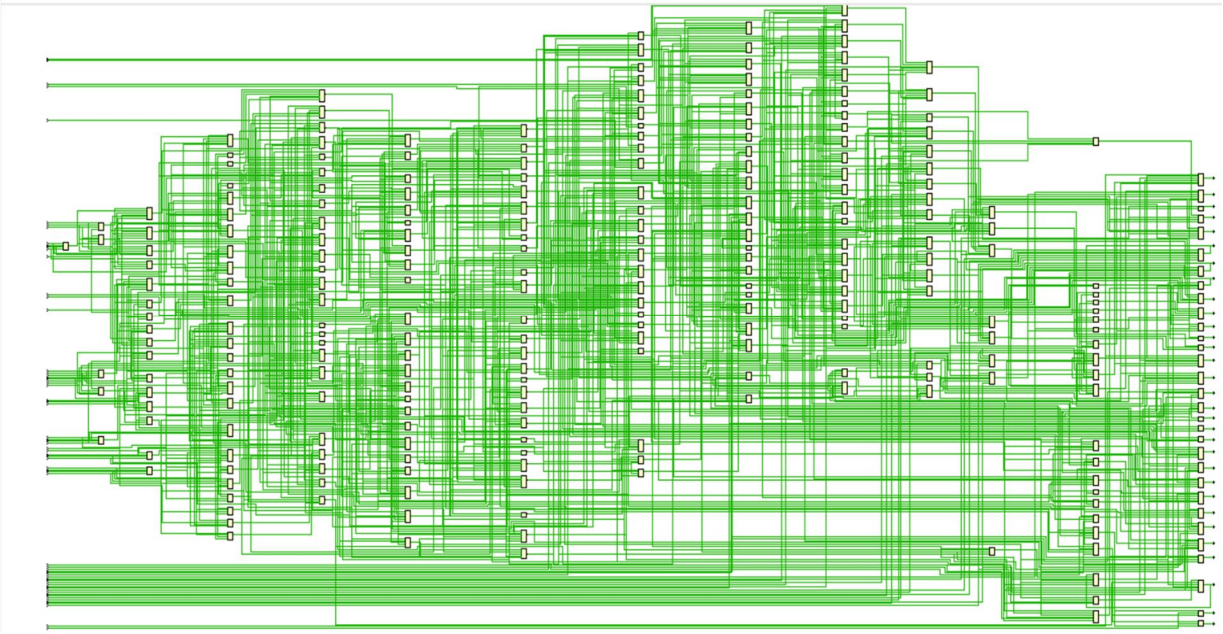


Fig 5. Technology Schematic of 32-bit Binary Polynomial Multiplier

C. Simulation Result

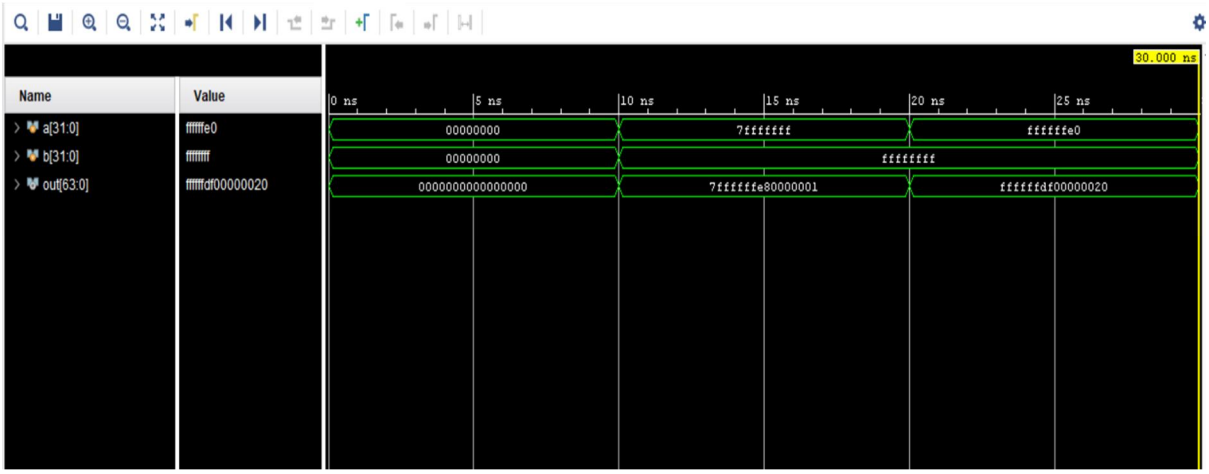


Fig 6. Simulation Waveform

D. Area

Name	Slice LUTs (134600)	Bonded IOB (400)
karatsuba32	1773	128

Fig 7. Area under Proposed method

E. Delay

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock
Path 1	∞	27	28	92	a[8]	out[59]	18.809	5.873	12.936	∞	input port clock	
Path 2	∞	27	28	92	a[8]	out[61]	18.809	5.873	12.936	∞	input port clock	
Path 3	∞	27	28	92	a[8]	out[60]	18.806	5.870	12.936	∞	input port clock	
Path 4	∞	27	28	92	a[8]	out[62]	18.806	5.870	12.936	∞	input port clock	
Path 5	∞	27	28	92	a[8]	out[63]	18.806	5.870	12.936	∞	input port clock	
Path 6	∞	26	27	92	a[8]	out[56]	18.329	5.768	12.561	∞	input port clock	
Path 7	∞	26	27	92	a[8]	out[58]	18.329	5.768	12.561	∞	input port clock	
Path 8	∞	26	27	92	a[8]	out[55]	18.326	5.765	12.561	∞	input port clock	
Path 9	∞	26	27	92	a[8]	out[57]	18.326	5.765	12.561	∞	input port clock	
Path 10	∞	25	26	92	a[8]	out[51]	17.849	5.663	12.186	∞	input port clock	

Fig 8. Delay under Proposed method

F. Comparison Table

	Area (LUT's)	Delay (ns)
Existing system	2074	25.732
Proposed method	1773	18.809

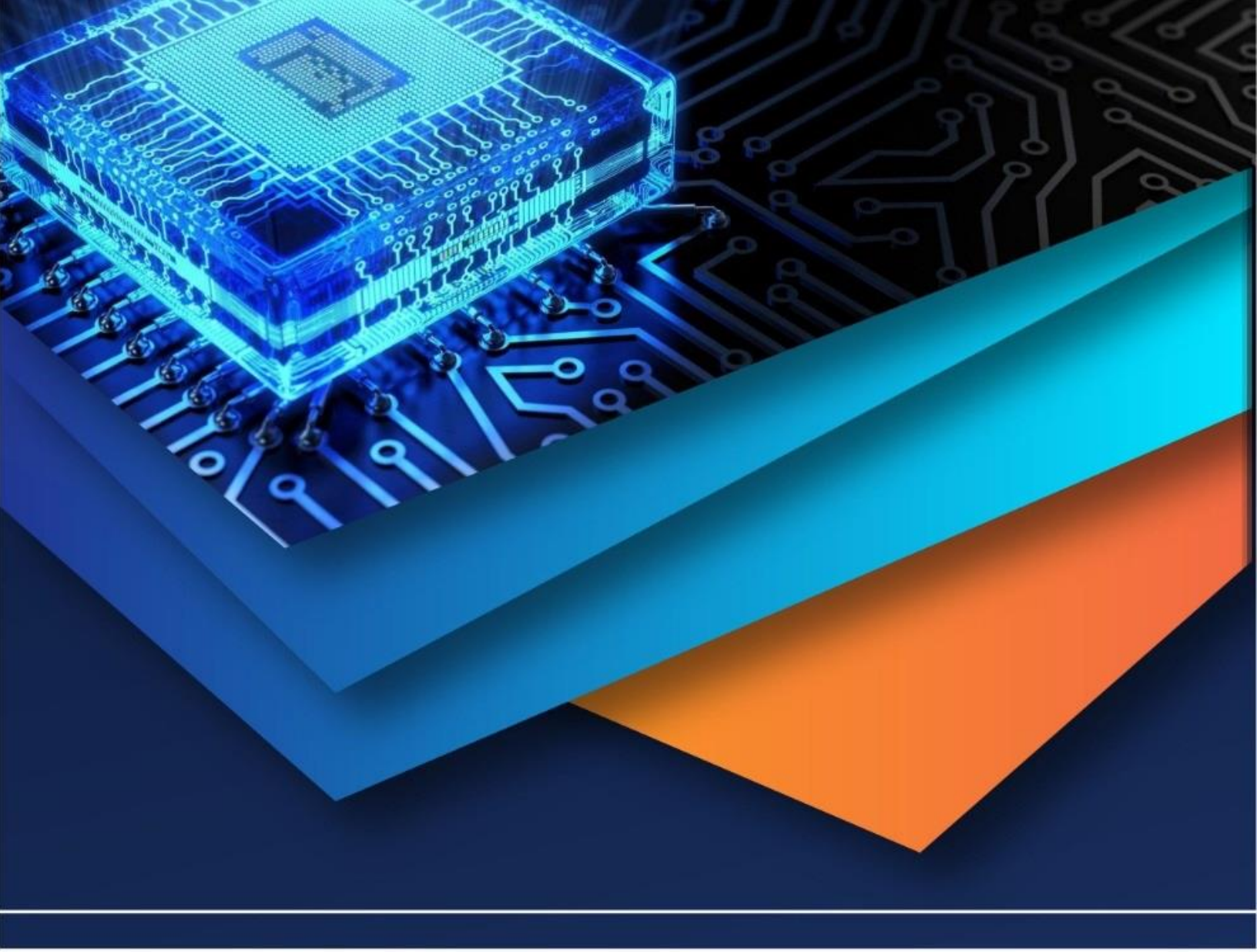
Table 1. Comparison between Existing system and Proposed method

VIII. CONCLUSION

In this article, 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). 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 (Wallace tree multiplier). The proposed method was extensively tested Xilinx to attain the improvement graph over other similar works. By comparing the area and the delay products of existing system and proposed system, the time and space complexities were drastically reduced. By this project, we had implemented a 32-bit Binary polynomial multiplier which are efficient in both area and delay values. This work achieved the suitable trade-off between space and time complexities, which minimizes the ADP requirement of the multiplier.

REFERENCES

[1] R. Abu-Salma, M. A. Sasse, J. Bonneau, A. Danilova, A. Naiakshina, and M. Smith, "Obstacles to the adoption of secure communication tools," in Proc. IEEE Symp. Secur. Privacy (SP), May 2017, pp. 137–153.
[2] B. Vembu, A. Navale, and S. Sadhasivan, "Creating secure communication channels between processing elements," U.S. Patent 9 589 159, Mar. 7, 2017.
[3] J. Yoo and J. H. Yi, "Code-based authentication scheme for light weight integrity checking of smart vehicles," IEEE Access, vol. 6, pp. 46731–46741, 2018.
[4] P. Aparna and P. V. V. Kishore, "Biometric-based efficient medical image watermarking in E-healthcare application," IET Image Process., vol. 13, no. 3, pp. 421–428, 2019.
[5] T. D. Premila Bai, K. M. Raj, and S. A. Rabara, "Elliptic curve cryptography based security framework for Internet of Things (IoT) enabled smart card," in Proc. World Congr. Comput. Commun. Technol. (WCCCT), Feb. 2017, pp. 43–46.
[6] Z. U. A. Khan and M. Benaissa, "High-speed and low-latency ECC processor implementation over GF(2m) on FPGA," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 1, pp. 165–176, Jan. 2017.
[7] G. Chen, G. Bai, and H. Chen, "A high-performance elliptic curve cryptographic processor for general curves over GF(p) based on a systolic arithmetic unit," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 54, no. 5, pp. 412–416, May 2007.
[8] H. Marzouqi, M. Al-Qutayri, K. Salah, D. Schinianakis, and T. Stouraitis, "A high-speed FPGA implementation of an RSD-based ECC processor," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 1, pp. 151–164, Jan. 2016.
[9] K. C. C. Loi and S. B. Ko, "Scalable elliptic curve cryptosystem FPGA processor for NIST prime curves," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 11, pp. 2753–2756, Jan. 2015.
[10] N. Y. Goshwe, "Data encryption and decryption using RSA algorithm in a network environment," Int. J. Comput. Sci. Netw. Secur. (IJCSNS), vol. 13, no. 7, p. 9, 2013.
[11] R. J. McEliece, Finite Fields for Computer Scientists and Engineers, vol. 23. Boston, MA, USA: Springer, 2012.
[12] N. Homma, K. Saito, and T. Aoki, "Toward formal design of practical cryptographic hardware based on Galois field arithmetic," IEEE Trans. Comput., vol. 63, no. 10, pp. 2604–2613, Oct. 2014.
[13] A. D. Piccoli, A. Visconti, and O. G. Rizzo, "Polynomial multiplication over binary finite fields: New upper bounds," J. Cryptograph. Eng., vol. 10, pp. 197–210, Apr. 2019.
[14] F. Mallouli, A. Hellal, N. Sharief Saeed, and F. Abdulraheem Alzahrani, "A survey on cryptography: Comparative study between RSA vs ECC algorithms, and RSA vs el-gamal algorithms," in Proc. 6th IEEE Int. Conf. Cyber Secur. Cloud Comput. (CSCloud)/5th IEEE Int. Conf. Edge Comput. Scalable Cloud (EdgeCom), Jun. 2019, pp. 173–176.
[15] B. Sunar, "A generalized method for constructing subquadratic complexity GF (2k) multipliers," IEEE Trans. Comput., vol. 53, no. 9, pp. 1097–1105, Sep. 2004.
[16] K.-W. Kim and J.-C. Jeon, "Polynomial basis multiplier using cellular systolic architecture," IETE J. Res., vol. 60, no. 2, pp. 194–199, 2014.
[17] C.-Y. Lee, C.-C. Chen, Y.-H. Chen, and E.-H. Lu, "Low-complexity bit-parallel systolic multipliers over GF(2m)," in Proc. IEEE Int. Conf. Syst., Man Cybern., vol. 2, Oct. 2006, pp. 1160–1165.
[18] W. Tan, A. Au, B. Aase, S. Aao, and Y. Lao, "An efficient polynomial multiplier architecture for the bootstrapping algorithm in a fully homomorphic encryption scheme," in Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS), Oct. 2019, pp. 85–90.
[19] W. Liu, S. Fan, A. Khalid, C. Rafferty, and M. O'Neill, "Optimized schoolbook polynomial multiplication for compact lattice-based cryptography on FPGA," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 27, no. 10, pp. 2459–2463, Oct. 2019.
[20] C. Rafferty, M. O'Neill, and N. Hanley, "Evaluation of large integer multiplication methods on hardware," IEEE Trans. Comput., vol. 66, no. 8, pp. 1369–1382, Aug. 2017.
[21] Z. Gu and S. Li, "A division-free Toom-Cook multiplication-based Montgomery modular multiplication," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 66, no. 8, pp. 1401–1405, Aug. 2019.
[22] J. Ding and S. Li, "A low-latency and low-cost Montgomery modular multiplier based on NLP multiplication," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 67, no. 7, pp. 1319–1323, Jul. 2020.
[23] D. Zoni, A. Galimberti, and W. Fornaciari, "Flexible and scalable FPGAoriented design of multipliers for large binary polynomials," IEEE Access, vol. 8, pp. 75809–75821, 2020.
[24] M. Langhammer and B. Pasca, "Efficient FPGA modular multiplication implementation," in Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays, Feb. 2021, pp. 217–223.
[25] K. Safiullah, J. Khalid, and S. Y. Ali, "High-speed FPGA implementation of full-word Montgomery multiplier for ECC applications," Microprocessors Microsyst., vol. 62, pp. 91–101, Oct. 2018.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY