International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 3, May-Jun-2026, ISSN (Online): 2395-566X

Maximizing Area and Power Efficiency with a Modified Karatsuba Multiplier for Cryptography Algorithms That Avoid Errors

Dr. A. Ranganayakulu[1], Mr. A. Prasad[2], M. Ramana Reddy[3], B. Ajanta Reddy[4], Dr. D. Satya Narayana[5]

[1]Professor & HOD, Department of ECE, Krishna Chaitanya Institute of Technology & Sciences, Markapur.
[2]Associate Professor, Department of ECE, Krishna Chaitanya Institute of Technology & Sciences, Markapur.
[3]Assistant Professor, Department of ECE, Krishna Chaitanya Institute of Technology & Sciences, Markapur.
[4]Assistant Professor, Department of ECE, Krishna Chaitanya Institute of Technology & Sciences, Markapur.
[5]Associate Professor, Department of ECE, Krishna Chaitanya Institute of Technology & Sciences, Markapur

Abstract- Using efficient finite field multipliers becomes vital in elliptic curve cryptography (ECC), where data security and authentication are critical. These multipliers do affect performance, however, because they use quite a lot of hardware resources. The Karatsuba algorithm and its variations are explored in this study as a means to enhance hardware efficiency on FPGA devices. Although performance is improved with the overlap-free Karatsuba algorithm. Problems with recombining intermediate findings cause them to add 20% mistakes. We present a modified Karatsuba method that can compute four key outputs for 2-bit inputs error-free to solve this problem. The revised design tested on Artix-7 FPGA and implemented in Verilog HDL, cuts power consumption by 73.95% and area utilization by 95% when compared to the original Karatsuba algorithm. Its overall efficiency is much improved while accuracy is guaranteed, despite a 3.22% increase in area and an 11.12% increase in power compared to the overlap-free version.

Keywords- Artix FPGA board, cryptography, karatsuba algorithm, polynomial, xilinx vivado tool.

I. INTRODUCTION

Communication devices, autonomous vehicles, the Internet of Things (IoT), healthcare, and many more fields rely on cryptography to authenticate users and protect sensitive data. Because there are several varieties of cryptography, such as public-key and symmetric-key cryptography. Secure information exchange is made possible via public-key cryptography, which relies on digital signatures and key setup. Some examples of public-key encryption algorithms include elliptic curve cryptography (ECC), Diffie-Hellman, RSA, and ElGamal. Because of its compact key sizes and robust security features, ECC is easy to deploy. Faster, cheaper, more efficient, and better suited to the low-power requirements of electronic devices is hardware-based encryption as compared to software-based cryptography. This leads to the execution of ECC cryptographic algorithms using field-programmable gate array (FPGA) implementations.

The use of finite-field operations is necessary for the computation of elliptic curve points. Because of its larger footprint, the finite-field multiplier has an impact on the algorithm's efficiency. A space-efficient multiplier based on the Karatsuba algorithm (KA) uses fewer multiplications and more addition operations. When it comes to n-bit conventional multipliers, KA only needs n1.58 for n2 single-digit multiplications. The repetitive nature of KA, on the other hand, compromises performance in exchange for temporal complexity. When applied to hardware, the Karatsuba algorithm suffers from severe power, area, and output constraints. Despite using fewer multiplications compared to naïve techniques, the output accuracy in some designs may be compromised due to the recursive structure's increased number of intermediate additions and subtractions. The algorithm's space requirements are higher due to the complexity of managing intermediate results and recombination’s, making it less efficient for systems with limited resources like FPGAs.

Even if there are speed benefits, the increased power consumption from running more processes and using more

memory might be too much for certain applications. When it comes to systems that prioritize energy efficiency and compactness, these drawbacks stand out even more. The spark has been lit, and now we can work on creating the Overlap free Karatsuba algorithm. By removing overlapping terms from intermediate calculations, the overlap-free Karatsuba algorithm achieves better output accuracy than its overlap counterpart. Applications that need accuracy, like encryption, may benefit from this approach's enhanced dependability. Nevertheless, there is a little increase in both space and power consumption due to the increased number of processes caused by the removal of overlaps. Although it improves accuracy, it may not be the best choice for systems with limited resources or those that are sensitive to power because of the trade-off between hardware resources and energy efficiency.

The modified Karatsuba algorithm improves upon the original Karatsuba method in order to make it more efficient and accurate. By guaranteeing accurate computations for critical outputs, it fixes mistakes made during intermediate result recombination. The updated version fixes the overlap-free variant's errors, particularly for tiny inputs, by treating the intermediate stages more carefully during multiplication. This maintains the algorithm's efficiency while producing error-free outputs, even with 2-bit inputs. In comparison to conventional Karatsuba algorithms, the updated version achieves a happy medium between speed and accuracy, making better use of hardware resources while decreasing power consumption and space. This is the extensive context from the prior literature reviews.

The area and power of the multipliers are now the primary foci of this work. Because of its recursive nature and the need to store intermediate results, the Karatsuba algorithm consumes a lot of hardware room and power, although it is efficient at decreasing multiplication operations. Due to its increased complexity, the overlap-free Karatsuba method uses somewhat more space and power than the normal Karatsuba algorithm, but it increases efficiency by removing unnecessary intermediary stages. However, optimizing for both area and power efficiency is achieved via the modified Karatsuba algorithm. In comparison to the original Karatsuba algorithm, it guarantees error-free outputs while reducing space utilization and power consumption.

This improved technique is ideal for hardware implementations with limited resources because, despite a little increase in area and power compared to the overlap-free version, it provides a better balance between efficiency, accuracy, and resource utilization. Encryption, digital signatures, and ECC are some of the modern cryptographic techniques used to protect data. It works using secure communications and blockchain technology. Emergence of quantum computing has prompted the creation of new algorithms to guarantee the safety of data in the future. The physical space needed to construct a circuit is called area in hardware design, whereas power is the amount of energy used by the circuit while it is operating. Both play an important role in hardware optimization, particularly in settings with limited resources, such as embedded systems and field-programmable gate arrays (FPGAs).

Devices may be made smaller and cheaper by minimizing their area, and they can be made more energy efficient and last longer on a single charge by decreasing their power consumption. To achieve peak performance without wasting resources, efficient designs try to find a happy medium between area and power. This is what drives the effort. Practically speaking, pipelines help the algorithm run faster. In prose, several algorithms and methods are suggested. In [1], the implementation of multistage ring oscillators was explored via the use of Look-Up Tables for rapid carry chains. With predictable routing and a high level of frequency sensitivity and PVT (process, voltage, temperature) tolerance, consistent routing methods are guaranteed, even with hardware slice occupancy of up to 50% and energy utilization of up to 44%. Optimal implementation of the Karatsuba method for asymmetric digit multipliers of ECC is carried out in [2].

Hardware efficiency may be enhanced by the use of parallel processing and pipelining techniques. The total speed is improved by the interleaved quick reduction techniques and pipelined modular multiplication. Side channel attacks may be better defended with countermeasures such as scalar blindness and base-point randomization. In [3], we look into Parallel Decimal Multiplier and how symmetric and asymmetric partitioning techniques impact the Karatsuba algorithm. In exchange for some extra space, these methods improve the Karatsuba algorithm and cut power usage by 25%. Additionally, they enhance power dissipation in the system by isolating switching operations. Because of this, these algorithms are ideal for mobile devices that are power-constrained.

In order to efficiently perform point multiplication (PM) and double point multiplication (DPM) for signature and verification operations in ECDSA, the Differential Addition Chain (DAC) was developed in [4]. This design is perfect for time-sensitive 5G applications since it lowers the area-time product and boosts throughput efficiency. Specifically for Shor's solution to the elliptic curve discrete algorithm problem (ECDLP) and binary point addition, Binary Elliptic Curves are used to optimize depth in order to decrease processing time for quantum cryptoanalysis in [5]. Through improvements in the Qiskit quantum computer simulator, it enhances the existing circuit implementation of the FLT-based inversion and Karatsuba multiplier.

Cutting down on the number of CNOT gates and the depth of the circuit improves efficiency. On top of that, this approach gets rid of 90% of the Toffoli gates needed for a one-step point addition. The Urdhva Triyagbhyam Sutra provides the basis for the high-speed Vedic multiplier that is invented in [6] and [7]. Nonetheless, the suggested FIR filter cuts down on area by 51.11%, latency by 45.83%, and power consumption by 19.49% [6]. This was achieved by using this technique. When it comes to accurate filtering, the suggested design beats integer-based FIR filters as the number of taps increases. Lowering the delay from 58.04% to 46.95% and the area from 58.72% to 49.77% are the specific reductions. Nevertheless, a cost-benefit analysis reveals that power dissipation rises from 45.41% to 47.50% [7].

To determine which ECC method is best for encryption and decryption, [8] compares many methods, including Rivest, Shamir, Adleman, etc. Alternative ways for implementing constrained-resource designs in multiple coordinate systems are given in [9]. These designs include things like point multiplication in encryption. The performance and efficiency of the Montgomery ladder point multiplication algorithm are examined using the Python programming language. In [10], the authors provide hardware architecture for variable-sized large integers that are flexible using Karatsuba multipliers. Optimizing and accommodating operands of various sizes in hardware implementation is achieved via the use of the recursive breakdown technique. When applied to a Xilinx Zynq MP FPGA, this method provides a 9.2× performance boost compared to well optimize software libraries. The article delves into an advanced approach to expand the scope of cloud data security utilizing ECC [11].

So that a comparable database transmitter and receiver aren't needed, the raw data and CII characters are mixed and delivered into ECC as a source. For elliptic curve cryptography (ECC) point multiplication (PM), [12] presents a dual-field architecture based on the Montgomery Ladder method that combines 6CC-6CC and 6CC-4CC designs. Optimal use of hardware resources allows for ECC methods to achieve great efficiency, and it is compatible with GF (2^283) and GF (2^571), among other binary fields. Faster processing and more efficient use of resources. It takes 17.44 µs and 12.55 µs, respectively, to execute a single PM operation on different FPGA systems. In [13], RSA cryptography is used to solve the discrete logarithm issue in small fields and the factorization decomposition problem for huge numbers.

The steady-state elliptic curve collecting process is investigated in detail, together with the strategy values of ECC for the important internal components of the system. In [14], we see the development of an innovative ECC-based high-security iris identification system that is tested for accuracy, privacy, and security utilizing the CASIA-IrisV3 database. An individual point represents one of the binary shards that make up the original Iris Code. A complex method for multiplying polynomials that has been optimized for quantum multiplication in order to maximize space and time efficiency is described in [15] as the Toom-Cook 8-way multiplier algorithm. With a Toffoli depth of O (n1.0569) and a smaller asymptotic qubit count of roughly O (n1.245), the Toom-Cook 8-way it is an efficient algorithm. It makes it harder to be hit by side-channel attacks like CPA, which use multiplication.

The characteristics of Hyper Elliptic Curve Cryptography (HECC), such as its compressibility, untraceability, finite condition, and key size, are studied and enhanced in [16]. By eliminating an XOR gate from the critical path, the overlap-free Karatsuba algorithm (OKA) enhances the performance of the classic KA. On the other hand, this multiplier is slower than the usual one, and the trade-off involves mistakes. In order to enhance speed and decrease calculation mistakes, this work proposes a modified Karatsuba algorithm that strikes a compromise between area and speed. It takes a lower-level base unit like KA and mixes it with a technique like OKA. Using polynomial multiplications for operand sizes ranging from small to large; all three versions are executed on FPGA. Part II takes a look at the Karatsuba algorithm, specifically the overlap-free version. Section III explains the revised Karatsuba algorithm.

In part IV, we cover the results of the simulation and the implementation, and then we get to the conclusion. Section II: Karatsuba Algorithms the Karatsuba Algorithm and variants were created to enhance efficiency compared to traditional multipliers. Both the 2-bit and 4-bit KA structures are shown in fig.1. One of the most important features of public-key cryptosystems such as RSA, ECC, and Diffie-Hellman is the ability to quickly multiply large integers. The Karatsuba method is used in cryptography for this purpose. One process with extremely huge numbers that these systems depend on is modular exponentiation, which comprises repeated multiplications. The naïve multiplication algorithm's temporal complexity is reduced to O (n2) using Karatsuba's divide-and-conquer technique, making it more efficient for large inputs. Faster multiplication is a performance boon for cryptographic algorithms, especially for uses requiring secure key generation and modular arithmetic.

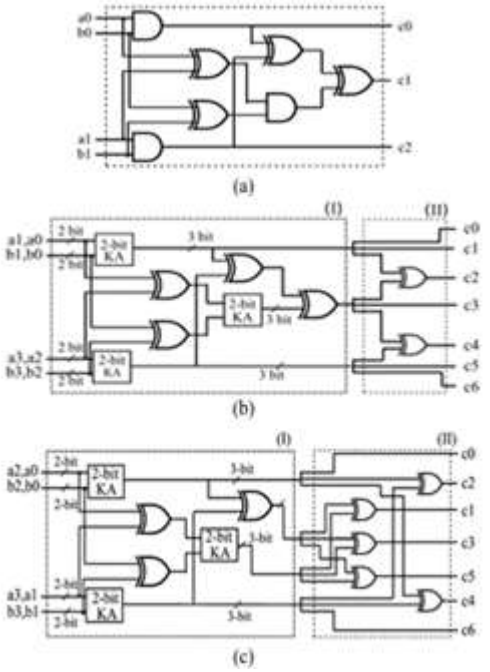


Fig. 1. Hardware implementation of (a) 2-bit KA and (b) 4-bit KA (c) 4-bit OKA

The calculation is carried out in the same way as seen in figure 1 for two polynomials of degree one, denoted as (=2), () = +, and () = +. Think about two n-term polynomials in (2) and ( ) for an n-bit multiplier. The following polynomials with a degree of -1 are given in equations (1) through (12).

$$CA_{XOR}(n) = (n-1)^2 \quad (1)$$

$$CA_{AND}(n) = (n)^2 \quad (2)$$

$$T_{CA}(2) = T_x + T_a \quad (3)$$

$$T_{CA}(n) = T_a + log_2(n)\, T_x \quad (4)$$

$$A(x) = \sum_{i=0}^{n-1} a_i x^i \quad (5)$$

$$B(x) = \sum_{i=0}^{n-1} b_i x^i \quad (6)$$

$$A(x) = x^m \sum_{i=0}^{m-1} a_{m+i}\, x^i + \sum_{i=0}^{m-1} a_m\, x^i = A_H\, x^m + A_L \quad (7)$$

$$B(x) = x^m \sum_{i=0}^{m-1} b_{m+i}\, x^i + \sum_{i=0}^{m-1} b_m\, x^i = B_H\, x^m + B_L \quad (8)$$

$$A(x)B(x) = (A_H\, x^m + A_L)(B_H\, x^m + B_L) \quad (9)$$

$$= P_2(x)x^{2m} + [P_1(x) - P_2(x) - P_0(x)]x^m + P_0(x) \quad (10)$$

$$P_2 = A_H B_H \quad (11)$$

$$P_1 = (A_H + A_L)(B_H + B_L) \quad (12)$$

The karatsuba algorithm multiplier for 2-bit input produces the equations (11) through (13) as its outputs. Then, using equations (13)–(19), we can determine the KA algorithm's area and space complexity.

$$P_0 = A_L B_L \quad (13)$$

$$KA_{XOR}(n) = 3\, KA_{XOR}\left(\frac{n}{2}\right) + 4n - 4 \quad (14)$$

$$KA_{AND}(n) = 3\, KA_{AND}\left(\frac{n}{2}\right) \quad (15)$$

$$T_{KA}(n) = 3\, T_X + T_{KA}\left(\frac{n}{2}\right) \quad (16)$$

$$KA_{XOR}(n) = 6\, n^{log_2(3)} - 8n + 2 \quad (17)$$

$$KA_{AND}(n) = n^{log_2(3)} \quad (18)$$

$$T_{KA}(n) = T_a + (3\, log_2(n) - 1)T_x \quad (19)$$

The space complexity in KA is reduced from quadratic (2 in the conventional technique) to subquadratic (2(3) = 1.58). In

contrast, an increase from 2( ) 3 2( ) is shown by the time complexity. Finally, KA decreases the area of the multipliers, but at the expense of a slower price. Consequently, the original Karatsuba is replaced with the overlap free KA (OKA), a variant designed for speed, to circumvent these issues (see fig. 1(c)). Instead of dividing inputs into high and low sections, they are separated into odd and even orders to optimize the longest route latency. It is assumed again that () and () are two polynomials in (2) and = 2. The following equation (20)–(27) is shown.

$$A(x) = \sum_{i=0}^{m-1} a_{2i}\, x^{2i} + \sum_{i=0}^{m-1} a_{2i+1}\, x^{2i+1} \quad (20)$$

$$B(x) = \sum_{i=0}^{m-1} b_{2i}\, x^{2i} + \sum_{i=0}^{m-1} b_{2i+1}\, x^{2i+1} \quad (21)$$

$$A(x) = \sum_{i=0}^{m-1} a_{2i}\, y^{i} + x \sum_{i=0}^{m-1} a_{2i+1}\, y^{i} = A_e(y) + xA_o(y) \quad (22)$$

$$B(x) = \sum_{i=0}^{m-1} b_{2i}\, y^{i} + x \sum_{i=0}^{m-1} b_{2i+1}\, y^{i} = B_e(y) + xB_o(y) \quad (23)$$

$$A(x)B(x) = \left(A_e(y) + x\,A_o(y)\right) \times \left(B_e(y) + x\,B_o(y)\right) = G_2(y)y + [G_1(y) - G_2(y) - G_0(y)]x + G_0(y) \quad (24)$$

$$G_0 = A_e B_e \quad (25)$$

$$G_1 = (A_o + A_e)(B_o + B_e) \quad (26)$$

$$G_2 = A_o B_o \quad (27)$$

It is possible to exclude an OR gate from the Karatsuba multiplier's critical path since the components of the odd and even terms do not overlap in this case. The results of the 2-bit input overlap-free method multiplier are given by equations (25), (26), and (27). Equation (28) (33) then shows the area and space complexity of the OKA algorithm.

$$OKA_{XOR}(n) = 3\, OKA_{XOR}\left(\frac{n}{2}\right) + 4n - 4 \quad (28)$$

$$OKA_{AND}(n) = 3\, OKA_{AND}\left(\frac{n}{2}\right) \quad (29)$$

$$T_{OKA}(n) = 2\, T_X + T_{OKA}\left(\frac{n}{2}\right) \quad (30)$$

$$OKA_{XOR}(n) = 6\, n^{log_2(3)} - 8n + 2 \quad (31)$$

$$OKA_{AND}(n) = n^{log_2(3)} \quad (32)$$

$$T_{OKA}(n) = T_a + (2\, log_2(n) - 1)T_x \quad (33)$$

We find that the 2-bit KA, according to our calculations, should provide four outputs, but there are only three accessible, which creates an error by definition. Therefore, we use the updated KA to create the KA that is free of errors. Section Three: Revised Karatsuba Algorithms Improving upon the original Karatsuba multiplication algorithm, the modified karatsuba algorithm yields better results. The time complexity is reduced from () to (), which is almost equivalent to (), in the traditional Karatsuba method. This is achieved by splitting two huge integers into smaller portions and performing three recursive multiplications instead of the four in classical multiplication. Possible enhancements to the updated version include optimizing base cases, lowering the amount of recursive calls, or adding more heuristics for dividing the numbers. These enhancements may enhance the algorithm's performance for big number multiplication by reducing the number of operations, optimizing memory use, or tailoring it to certain number types or hardware architectures. The modified KA is used for multiplication as indicated in fig.2, since error-free smaller blocks are necessary for efficient and safe ECC. If we take the same polynomial form into consideration, then

$$C0 = a0\,.\,b0 \quad (34)$$

$$C1 = (a0\,.\,b1) + (a1\,.\,b0) \quad (35)$$

$$C2 = a1\,.\,b1\,.\,(a0.\,b0) \quad (36)$$

$$C3 = a1\,.\,b1 \quad (37)$$

Results from the 2 bit input multiplier using the modified Karatsuba method are given by equations (34), (35), (36), and (37) respectively. The OKA algorithm's area and space complexity are then determined using the formulas in equations (38)–(43).

$$KA_{XOR}(n) = 3\, KA_{XOR}\left(\frac{n}{2}\right) + 4n - 4 \quad (38)$$

$$KA_{AND}(n) = 5\, KA_{AND}\left(\frac{n}{2}\right) \quad (39)$$

$$T_{KA}(n) = 3\, T_X + T_{KA}\left(\frac{n}{2}\right) + 1\, KA_{NOT} \quad (40)$$

$$KA_{XOR}(n) = 6\, n^{log_2(3)} - 8n + 2 \quad (41)$$

$$KA_{AND}(n) = n^{log_2(5)} \quad (42)$$

$$T_{KA}(n) = T_a + (3\, log_2(n) - 1)T_x. \quad (43)$$

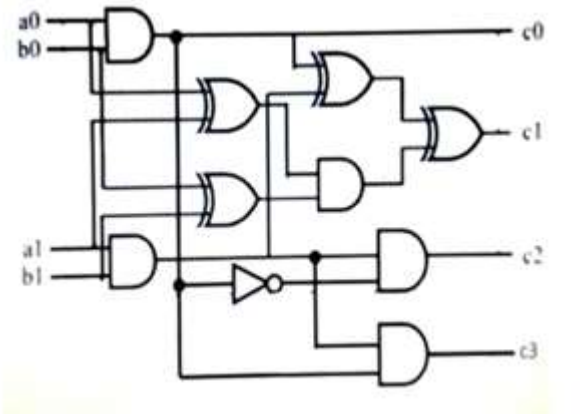


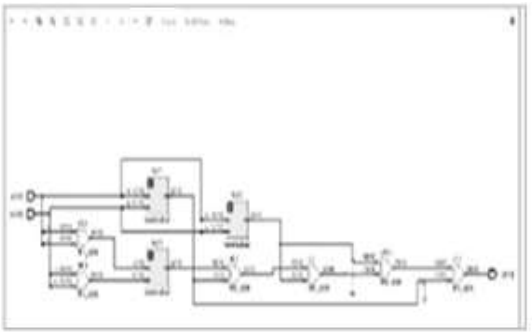
Fig. 2. Hardware implementation of 2-bit Modified KA

II. RESULTS AND DISCUSSION

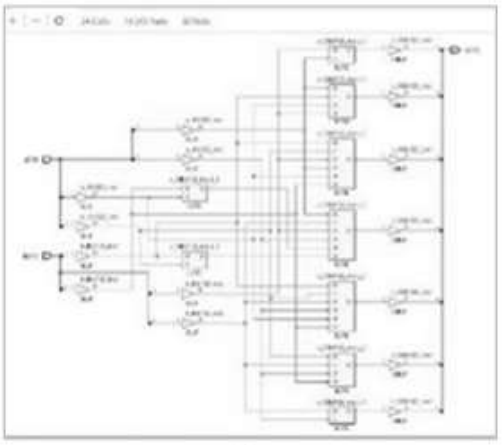
Xilinx Vivado is used to design the multipliers for the Artix-7 FPGA board, which is based on 28nm CMOS technology (XC7A200TLFFG1156-2L). The minimum system requirements for Xilinx Vivado 2023 are Windows 10, 4 GB of RAM, and an operating system.2. The developed multipliers' simulation results are given in figure 3, with a and b serving as inputs and c as the output.

Fig. 3. Simulation Result of 4-bit KA

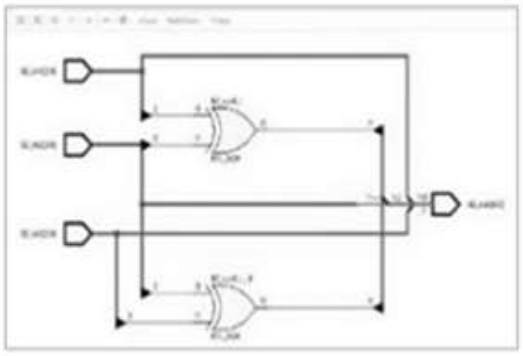
For 4-bit KA, OKA, and MKA, the RTL and technological schematics are as shown in figure 4. Even though it uses the same block several times, the OKA's RTL schematic is the simplest. The look-up tables and I/O buffers are used in the technological schematic. For MKA, the technological diagram is the most basic.



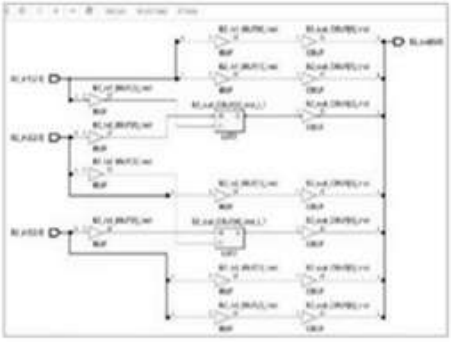
(a)



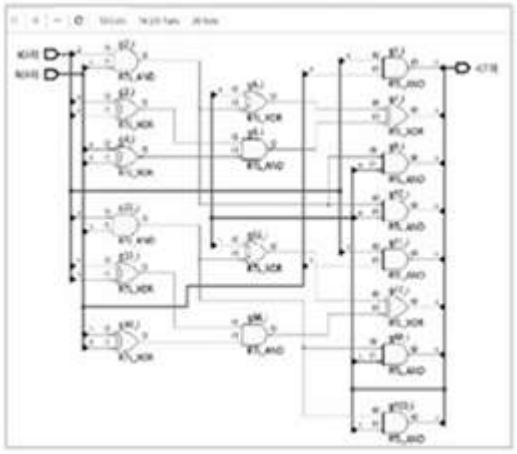
(b)



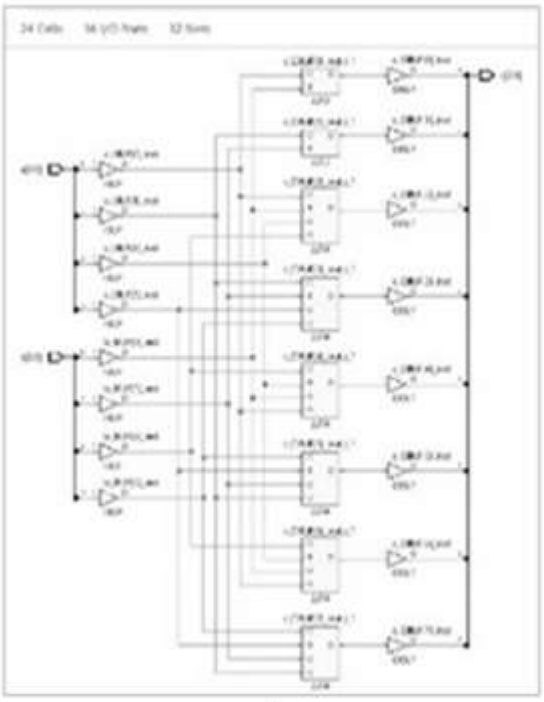
(c)



(d)



(e)



(f)

Fig. 4. RTL and Technology Schematics of 4-bit KA, OKA and MKA (a) RTL Schematic of 4-bit KA (b) Technology Schematic of 4-bit KA (c) RTL Schematic of 4-bit OKA (d) Technology Schematic of 4-bit OKA (e) RTL Schematic of 4-bit MKA (f) Technology Schematic of 4-bitM KA

Table I displays the outcomes of the synthesis and implementation. As the amount of input bits increases, so do the Slice LUTs. When comparing the MKA to the KA and OKA, the slice LUTs are reduced by 95.35% and -3.22%, respectively, as shown in Table I. When compared to KA and OKA, the MKA significantly lowers on-chip power by 83.88% and 24.89%, respectively. Therefore, rather than progress, the negative sign signals degeneration. All three algorithms—KA, OKA, and MKA—are thoroughly tested for introduced mistakes. Together, KA and OKA introduce 20% and 60% of the mistakes, respectively. The operation of the MKA is determined to be error free. The Modified Karatsuba Algorithm (MKA) is recursive; hence its dynamic power consumption grows as the input bit width and polynomial degree do. There are more sub-problems and operations (such addition, subtraction, and multiplication) that result in more recursive calls as the input bit-width rises. Because the method is based on reducing huge numbers to smaller ones, increasing the bit-width leads to more recursive levels, which raises the power consumption because each level requires more operations. Similarly, the complexity and computing resources needed to multiply the polynomial expand in direct proportion to its degree, as the number of terms to be multiplied increases. The upshot is increased dynamic power consumption due to increased clock cycles and switching activity. As the input size grows, both of these things add up to a dramatic increase in power consumption.

Table1 COMPARISON IN TERMS OF AREA AND POWER DISSIPATION

III. CONCLUSION

Because of their larger footprint and impact on algorithm performance, finite field multipliers are reserved for elliptic curve cryptography applications. This is remedied by increasing hardware efficiency using the Karatsuba algorithm or one of its variations. Modeled in Verilog HDL and tested on Artix-7 FPGA are the Karatsuba, overlap free Karatsuba, and Modified Karatsuba algorithms. Findings show that compared to the overlap-free Karatsuba algorithm, the space required for slice LUTs and power dissipation by the Modified Karatsuba Algorithm is at least 95% smaller and 73.95% smaller, respectively, whereas the overlap-free approach requires 3.22%

more space and 11.12% less power. The suggested method has room for improvement in terms of both accuracy and area reduction in the future.

REFERENCES

1. F. Spagnolo, S. Perri, F. Frustaci, F. Crupi, M. Vatalaro, and P. In IEEE Transactions on Very Large Scale Integration (VLSI) Systems,volume, Corsonello explains "Exploring the Usage of Fast Carry Chains to Implement Multistage Ring Oscillators on FPGAs: Design and Characterisation." August 2024; 32, no. 8, pp. 1472-1484; doi: 10.1109/TVLSI.2024.3395302.
2. A. J. Park, M. Awaludin, R. W. Wardhani, and H. Kim, 2009. "A High Performance ECC Processor Over Curve448 Based on a Novel Variant of the Karatsuba Formula for Asymmetric Digit Multiplier," in the IEEE Access, vol. 10, 20, 22; doi: 10.1109/ACCESS.2022.3184786, pp. 67470-67481.
3. S. Gorgin, Nejad, H. Z., and J. "A Practical Energy/Power Reduction Approach for Parallel Decimal Multiplier," by A. Lee, IEEE Access, vol. doi: 10.1109/ACCESS.2022.3145001. 10, pp. 11372-11381, 2022.
4. X. He and colleagues, "A Universal Architecture for Single and Double Point Multiplications for ECDSA Based on Differential Addition Chains," in IEEE Access, vol. Publication number: 10.1109/ACCESS.2024.3390244, pages. 55434-55447, 2024.
5. D. H. T. Larasati, J. Ji, H. Putranto, R. W. Wardhani, and H. Kim "Depth-Optimization of Binary Elliptic Curves for Quantum Cryptanalysis," in IEEE Access, vol. 11, 2023; doi: 10.1109/ACCESS.2023.3273601. Pages. 45083-45097.
6. Satyam, Neelima K, M. Sandhiya, C. Padma, Shaik Jaffar Ali and Kumar Raja Meruva, "High Speed Single Precision 64-Tap FIR Filter Using Urdhva Tiryagbhyam Sutra," 2024 IEEE Students Conference on Engineering and Systems (SCES), Prayagraj, India, 2024, pp. 1-5, doi: 10.1109/SCES61914.2024.10652418.
7. Neelima K, H. Yogananda Reddy, G. Bhaskar, N. Mani Teja and N. K. Priya, "Design and Evaluation of 32-Bit N-Tap FIR Filter for Audio Processing Applications," 2024 1st International Conference on Innovative Sustainable Technologies for Energy, Mechatronics, and Smart Systems (ISTEMS), Dehradun, India, 2024, pp. 1-6, doi: 10.1109/ISTEMS60181.2024.10560173.
8. K. Shah, A. Bhadauria, P. Thakkar, J. Shah and H. Kaur, "Advancements in Elliptic Curve Cryptography: A Review of Theory and Applications," 2024 Parul International Conference on Engineering and Technology (PICET), Vadodara, India, 2024, pp. 1-6, doi: 10.1109/PICET60765.2024.10716041.
9. N. H. Sabbry and A. Levina, "Elliptic Curve Cryptography on Constrained Devices: A Comparative Study of Point Multiplication Methods," 2024 13th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 2024, pp. 1-5, doi: 10.1109/MECO62516.2024.10577953.
10. B. H and Kang. Huang, "FlexKA: A Flexible Karatsuba Multiplier Hardware Architecture for Variable-Sized Large Integers," in IEEE Access, volume Publication 10.1109/ACCESS.2023.3282646, pages 55212–55222, 2023. Nu
11. V. Srinadh, B. Maram and T. Daniya, "Data Security And Recovery Approach Using Elliptic Curve Cryptography," 2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), Bangalore, India, 2021, pp. 1-6, doi: 10.1109/CSITSS54238.2021.9683473.
12. J. Li, Y. Luo, W. Wang, J. Zhang, and S. Chen, "Innovative Dual Binary-Field Architecture for Point Multiplication of Elliptic Curve Cryptography," in IEEE Access, volume In 2021, volume 9, pages 12405–12419, doi: 10.1109/ACCESS.2021.3051282.
13. J. VenkataGiri and A. Murty, "Elliptical Curve Cryptography Design Principles," 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 2021, 10.1109/RTEICT52294.2021.9573662. pp. 889-893, doi:
14. A. A. Asaker, Z. F. Elsharkawy, S. Nassar, N. Ayad, O. Zahran and F. E. Abd El-Samie, "A Novel Iris Cryptosystem Using Elliptic Curve Cryptography," 2021 9th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC), Alexandria, Egypt, 2021, pp. 155-158, doi: 10.1109/JAC ECC54461.2021.9691307.
15. D. H. T. Larasati, H. C. Putranto, R. W. Wardhani, and H. "Space and Time-Efficient Quantum Multiplier in Post Quantum Cryptography Era," by Kim, IEEE Access, vol. doi: 10.1109/ACCESS.2023.3252504; 11, pp. 21848-21862, 2023.

16. A. Yadav, P. Sharma and Y. Gigras, "A Comparative Study of Elliptic curve and Hyperelliptic Curve Cryptography Methods and an Overview of Their Applications," 2024 International Conference on Intelligent Systems for Cybersecurity (ISCS), Gurugram, India, 2024, pp. 01-06, doi: 10.1109/ISCS61804.2024.10581015.