JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A Survey on Different Methodologies of Modular Multiplication

Very Large Scale Integration

[1]Nikita S. Sukhadeve, [2]A.M.Shah
[1]Master of Technology Student, [2]Assistent Professor
[1]Department of Electronics Engineering,
[1]Government College of Engineering, Amravati, Maharashtra, India

Abstract : The demand of the decade is for low area, low power, and efficient arithmetic operations. The mathematical function of multiplication must be carried out quickly and efficiently in high-speed systems like digital signal processing, image processing, graphics, etc. When developing a system with complicated operations, it's crucial to take into account a number of critical factors, including the portability of a device, power consumption, the system's response time, and power dissipation. Modular operations are used in many of the algorithms used in applications like cryptography, which is the act of classifying information, changing it to a form that may be unintelligible, and protecting it from unauthorized persons. The length of the output in modular operations is constant, unlike other arithmetic operations. Partially produced products are generated in the first stage of multipliers, which is nothing more than a series of AND gates. The incomplete items are then combined to produce the complete products. Taking all of this into account takes time. This paper reviews the most popular and cutting-edge approaches for effective modular multiplication in this study, looking specifically at FPGA-based solutions while also assessing their advantages and disadvantages.

***IndexTerms* - Modular Multiplication, Karatsuba Algorithm, Cryptosystem.**

I. Introduction

The use of electronic communication is rapidly expanding, and information security concerns should follow suit [17]. Data sent across open computer networks need to be verified, kept private, and have its integrity guarded against tampering. Electronic businesses need digital valid signatures and secure payment methods to operate properly. All of these issues, as well as numerous more, are resolved by cryptography [22]. Providing confidentiality, a service used to keep publicly available information secret from everyone but those with access rights is one of cryptography's fundamental goals. There are various techniques to guarantee confidentiality. They range from physical defense to mathematical solutions that obfuscate the data. It employs the encryption and decryption techniques [17], [22], [19], and [20]. Many public-key cryptosystems, including Diffie and Hellman [28], [25], and the Rivest, Shamir, and Adleman encryption schemes [27], use modular exponentiation as a standard operation for scrambling. The three components of the RSA cryptosystem are a modulus M of around 1024 bits and two numbers, D and E, which are referred to as the private and public keys and meet the property TDE≡T mod M. T obeying $0 \leqslant T < M$ in plain text. Using the public key, messages are encrypted as C = TE mod M and are uniquely decrypted as T = CD mod M. Encryption and decryption are thus carried out using the same operation. It is decided that the modulus M will be the product of two enormous prime numbers, let's say P and Q. The encryption stage is normally quick since the public key E is typically tiny and only has a few bits set (i.e., bits = 1). The private key D is chosen so that DE = 1 mod (P-1)(Q-1) and has the same number of bits as the modulus M. Due to the difficulty of computationally discovering P and Q, the system is secure. It has been demonstrated that an RSA cryptosystem with a modulus of 1024 bits or higher cannot be cracked. Repeatedly applying modular multiplication is known as modular exponentiation. Therefore, the modular multiplication and exponentiation's implementation efficiency play a vital role in determining how well public key cryptosystems function. It is crucial to try to reduce the number of modular multiplications performed and the time required by a single modular multiplication because the operands (the plaintext, the cipher text, or possibly a partially ciphered text) are typically large (i.e. 1024 bits or more). This will improve the time requirements of the encryption/decryption operations.

It is possible to execute modular multiplication A×B mod M in one of two ways: first multiplying, or computing P = A×B, and then reducing, or computing R = P mod M, or by interspersing the multiplication and reduction phases. Modular multiplication is implemented via many algorithms. The most well-known ones are interleaving multiplication and reduction, Barrett's [24], [23], [21], and Karatsuba's [26] method for multiplying. This is how the review will be structured.

.

II. Literature Review

The digital World is considered Supremacy in the expanding sector of the world which requires an outrageous amount of power to run it. Hence, the more the power the unbeatable it becomes. For upholding the power in digitization, the Multiplication operation and the Vedic approach are used. But Multiplication is considered to be complex to design and the speed of multiplication is foremost. Hence, to increase efficiency, decrease expenditure and lessen the delay time the simplest algorithm which is Karatsuba Algorithm is used. Various types of multiplications schemes are implemented based on the FPGA. So, many researchers have performed on different types of architecture and analyzed or study various types of topologies which are summarised below:

Xinmiao Zhang, et.al. [1] suggested that a method known as the ring-learning with errors (R-LWR) could be used to create a variety of ciphers that are resistant to quantum computing assaults and fully homomorphic encryption, which enables computations to be performed on encrypted data. The modular reduction is to be incorporated into the Karatsuba polynomial multiplication using a novel method that is proposed in this research. The ultimate product is not subject to modular reduction; rather, it applies to goods from the intermediate segment. As a result, increased substructure sharing is made possible and a significant decrease in the number of coefficient additions is required for assembling the segment products to produce the desired outcome. The proposed method decreases the number of adds by 13–17% for polynomial multiplications with decomposition factors 2, 3, and 4.

Ashly George, et.al. [2] used three ones of half-length operands in parallel, the Karatsuba-Ofman multiplier substitutes for multiplication. In many public key cryptosystems, including RSA, modular multiplication is a fundamental operation that must be carried out. In terms of delay, area, and power, the suggested design exhibits higher performance.

Shankar R, et.al. [4] presented that Splitting the operands into two equal-length portions allows the Karatsuba algorithm to be started, which speeds up the multiplication of huge numbers. The carry propagation from LSB to MSB is decreased by the Vedic multipliers' creation of partial products and sums in a limited number of steps. The area and the latency are minimized in the suggested design. Finally, the outcomes of the Vedic and Karatsuba multipliers are compared.

Kiran Kumar V G, et.al. [5] offered a technique for using effective arithmetic algorithms, which serve as the foundation for intricate activities like signal and image processing and DSP. The addition, multiplication, and modular operations are all arithmetic operations. Reversible gates and the Vedic approach are combined to act as multipliers and are examined. With the various multipliers and the modular reduction methods used here, Montgomery's modular operation is adjusted, and the effectiveness of the modification is tested. The algorithms' area, timing, and power usage are tallied and investigated. The design's LUTs, slice registers, and IOBs are tabulated. The tabulated data assist the designer in making an effective algorithm selection based on the resources available during design. So an algorithm may be application-specific. Xilinx 14.2's Spartan 6 family and Cadence's 45nm technology are both used to implement all of the algorithms. Verilog is the chosen language for hardware description.

Mishal Jasmine Ferrao, et.al. [6] studied on three modular reduction techniques and one modular multiplication algorithm and its implementation.

Kumm, M., et.al. [7] worked on expands of Karatsuba's strategy for effectively using rectangular multipliers as the foundation for larger multiplies. The suggested method reduces resource usage and boosts efficiency for multipliers of integers larger than 64 bits.

Pasluri Bindu Swetha, et.al. [8] proposed an exportable application-specific direction set elliptic bent cryptography processor with a focus on repeating marked digit depiction. To achieve high throughput augmentation, the processor uses broad pipelining algorithms for the Karatsuba-Ofman strategy. Using Xilinx 13.2, the proposed design of this article investigates the reasoning for size, region, and power usage. Vedic Sutra - Nikhilam Sutra is the task's growth.

Arish, S., et.al. [9] propound a study of a highly effective run-time-configurable floating-point multiplier for matrix element multiplication along with an effective Strassen's algorithm for matrix multiplication. and The binary multiplier is implemented using a highly effective mix of the Urdhva Tiryagbhyam algorithm and the Karatsuba algorithm. By reconstructing itself while running, this design may efficiently change the power and delay needs in accordance with various accuracy requirements.

Can Eyupoglu, et.al. [11] put forward One of the algorithms created to increase effectiveness and decrease cost in order to simplify multiplication is the Karatsuba algorithm. The effectiveness of the Karatsuba algorithm is examined in this study in terms of the number of multiplications and the overall processing time for various bit lengths.

Sunil Devidas Bobade, et.al. [12] suggested an area-optimized, low-latency multiplier for use in ECC design, which performs the effective KOA method in a completely new way. The suggested algorithm employs a novel method of separating input operands according to exponent's parity, which ultimately aids in lowering the FPGA footprint and provides low latency by avoiding overlapping, a major consideration for any embedded system. They looked into how much space the suggested multiplier and cryptoprocessor occupied, and they came to the conclusion that the suggested scheme uses a lot less FPGA space than the one that uses a conventional KOA multiplier.

Shahram Jahani, et.al. [13] provide brand-new symbols in this study that were taken from the Big-ones binary representation of integers. To enhance the performance of big integer multiplication and squaring in number theory-based cryptosystems, they provide a modified version of the traditional multiplication and squaring algorithms based on the Big-ones. The suggested squaring algorithm is 2 to 3.7 and 7.9 to 2.5 times faster for squaring 32-bit and 8-Kbit values, respectively, than the widely used classical and Karatsuba multiplication procedures. Additionally, for multiplying 32-bit and 8-Kbit values, respectively, the suggested multiplication technique is 2.3 to 3.9 and 7 to 2.4 times faster. Since multiplication and squaring are the primary operations in the majority of these systems, the suggested technique directly benefits number theory-based cryptosystems that operate in the range of 1-Kbit to 4-Kbit integers.

Shri Prakash Dwivedi, et.al. [14] presented a technique to multiply two binary values effectively and furthermore presented the Nikhilam method of Vedic mathematics as the basis for an integer multiplication algorithm.

Gary C.T. Chow, et.al. [15] have argued that a crucial component of cryptographic algorithms is the modular multiplication of long numbers. Although a number of FPGA accelerators for massive modular multiplication have been developed, earlier systems relied on O(N2) techniques. In this study, we provide a Montgomery multiplier that uses the faster O(N(log 3/log 2) Karatsuba algorithm.

Sameh M. Shohdy, et.al. [16] look into the implementation of an elliptic curve-based public key cryptosystem that performs better thanks to the effectiveness of the core Galois field arithmetic. If the binary Karatsuba multiplier is truncated at the n-bit multiplicand level and uses an effective classic multiplier method, it is more effective. This work can compute the GF(2191) multiplication in 45.889 ns.

N. Nedjah, et.al. [18] suggested to implement software and hardware efficiently. Nevertheless, the findings are dispersed throughout the literature. In this work, they reviewed the most popular and up-to-date techniques for effective modular multiplication, looking into and analyzing their pros and cons. They give a suitable hardware implementation for each of the methods.

Bewick, G.W., et.al. [23] proposed a thesis to find the quickest technique to implement binary multiplication. The multiplication method of Booth has been extended to represent products in a partially redundant form (redundant Booth). In terms of layout space, power, and delay, conventional Booth encoded multipliers outperform other techniques.

Barrett, P., et.al. [24] look into a description of the procedures used at Oxford University to put "off-the-shelf" digital signal processing devices to use in a high-speed implementation of the RSA encryption method. On a first-generation DSP, encrypting took an average amount of time (for 512-bit exponent and modulus).

ElGamal, T., et.al. [25] studied an expansion of the Diffie-Hellman key distribution mechanism to create a public key cryptosystem.

Rivest, R., et.al. [27] put forward an approach to disclosing an encryption key in public that does not also reveal the associated decryption key. This has two significant effects: 1) The communication can only be decoded by him because only he is aware of both the encryption and decryption keys. 2) A signer cannot later contest the authenticity of his signature, and signatures cannot be falsified. The difficulty of factoring the divisor n contributes to the system's security.

W. Diffie, et.al. [28] advanced one of the most crucial areas of computer science which is cryptography, and new types of cryptographic systems are required as a result of applications that reduce the requirement for safe key distribution methods. This paper explains how theories of communication and computation are starting to offer the means to resolve long-standing cryptography issues.

III. Modular Arithmetic

An integer-based arithmetic system that takes the remainder into account is called modular arithmetic. In modular arithmetic, numbers "wrap around" to leave a remainder when they reach a predetermined fixed amount (the modulus). As seen in Wilson's theorem, Lucas' theorem, and Hensel's lemma, modular arithmetic is frequently connected to prime numbers and is frequently used in computer algebra, computer science, and cryptography.

With a 12-hour clock, modular arithmetic can be used in an intuitive way. If the time presently is 10:00, the clock will display 3:00 rather than 15:00 in 5 hours. 15 minus 3, with a modulus of 12, equals 3.

IV. Introduction To Modular Math

The equation that results from dividing two integers is as follows:
a/b = q reminder r
The dividend is a.
b is the factor.
The quotient is q.
r is the remainder.
Sometimes, when dividing a by b, we are simply concerned with the leftover.
The modulo operator is an operator that can be used in certain situations (abbreviated as mod).
The identical a, b, q, and r as before would have led to - r = a mod b

This might be expressed as r = a modulo b. where the modulus b is referred to.
Eg 14/5 = 2 reminder 4 i.e. 14 mod 5 = 4.
.

Multiplication Properties

Numerous branches of mathematics use modular multiplication, which has a wide range of applications in areas including computer science, computer algebra, and cryptography.

Modular arithmetic's multiplication properties

a. If A . B = C, then A (mod n) . B (mod n) = C (mod n).

b. If A ≡ B (mod n), then kA ≡ kB (mod n) any k-th integer.

c. If A ≡ B (mod n) and C ≡ D (mod n), then AC ≡ BD (mod n).

V. Summary

We summarized some methodologies below:

Table 1 Summary

Sr. No.	Summary about Articles		
	Author Name	**Methodology**	**Finding**
1	Xinmiao Zhang, et.al. [1]	In this study, a novel approach is put forth for incorporating modular reduction into the multiplication of Karatsuba polynomials. Instead of the final product, the modular reduction is used on items from the intermediate segment.	Integrates the Modular Reduction into the Karatsuba Multiplication.
2	Kiran Kumar V G, et.al. [5]	Reversible gates and Vedic approach are combined to act as multipliers and are examined. With the various multipliers and the modular reduction methods used here, Montgomery's modular operation is adjusted, and the effectiveness of the modification is tested.	Multiplication Algorithms like Karatsuba Multipliers need to be compared with the Reversible Multipliers and Vedic Methodology to Determine the most effective and efficient Multiplier.
3	Gary C.T. Chow, et.al. [15]	In this paper, they describe a Montgomery multiplier that uses the faster O(N (log 3/ log 2) Karatsuba algorithm.	Karatsuba-Based Montgomery Multiplier is used for Cryptography Applications using long Integers.
4	Ashly George, et.al. [2]	Montgomery modular multiplication's most recent method, which uses the KO algorithm and can use less hardware. The proposed design outperforms existing ones in terms of delay, area, and power, and there has been a notable improvement in latency and the trade-off between area and performance.	Montgomery Modular Multiplier using Karatsuba-Ofman Algorithm is used in order to minimize the delay, area, and power.
5	Shankar R, et.al. [4]	The area and the latency are minimized in the suggested design. Finally, the outcomes of the Vedic and Karatsuba multipliers are compared.	The combination of Karatsuba-Vedic Multiplier helps to effective delay and the area of the circuit is thus reduced.

VI. Conclusion

This paper reviewed the most well-liked and most recent approaches to effective modular multiplication. In modular multiplication designs, the multipliers have much lower area-delay products. They deliver outstanding performance and energy efficiency as well. We described two methods for performing the modular multiplication A×B mod M: acquiring the product and then reducing it, or receiving the reduced product directly. Modular multiplication is implemented via many algorithms.

References

[1] Xinmiao Zhang, Keshab K. Parhi, “Reduced-Complexity Modular Polynomial Multiplication for R-LWE Cryptosystems”, in ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6-11 June 2021.

[2] Ashly George, Ajeesh S., Sindhu T.V., “Performance Analysis of Montgomery Modular Multiplier Using Karatsuba Algorithm”, in April-2021 International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering, Issue 4, April 2021.

[3] L´aszl´o Babai, “Divide and Conquer: The Karatsuba algorithm”, Updated 01-13-2021.

[4] Shankar R, Sundhararajan G, Vignesh S, Aravind AR., “FPGA Implementation of Karatsuba Vedic Multipliers”, in 2021 International Journal of Advance Research and Innovative Ideas in Education, 2021.

[5] Kiran Kumar V G1, Shantharama Rai C2, “Design and Implementation of Efficient Cryptographic Arithmetic Based on Reversible Logic and Vedic Mathematics”, in March -April 2020 International Journal of Advanced Trends in Computer Science and Engineering, 2020.

[6] Mishal Jasmine Ferrao1, Mr. Kiran Kumar. V. G2, Mrs. Megha N3, "Implementation of Modular Reduction and Modular Multiplication Algorithms", in IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) (Nov. - Dec. 2018), 2019.

[7] Kumm, M., Gustafsson, O., de Dinechin, F., Kappauf, J., Zipf, P., (2018), "Karatsuba with Rectangular Multipliers for FPGAs", 2018 IEEE 25TH SYMPOSIUM ON COMPUTER ARITHMETIC (ARITH),2018.

[8] Pasluri Bindu Swetha 1*, V.J. Kishore Sonti 2, A. Murali 3, "VLSI design for efficient RSD-Based ECC processor using Karatsuba algorithm", in International Journal of Engineering & Technology, (2018).

[9] Arish, S. and Sharma, R.K., "Run-Time-Reconfigurable Multi-Precision Floating-Point Matrix Multiplier Intellectual Property Core on FPGA", Circuits, Systems, and Signal Processing, 2017, pp. 998-1026, 2017.

[10] Haoyuan Sun, "Fast Multiplication: Karatsuba and FFT", May 2016.

[11] Can Eyupoglu*, "Performance Analysis of Karatsuba Multiplication Algorithm for Different Bit Lengths", in World Conference on Technology, Innovation and Entrepreneurship Published by Elsevier Ltd., Procedia - Social and Behavioral Sciences 195 (2015), 2015.

[12] Sunil Devidas Bobade1*and Vijay R. Mankar2, "Area Optimized Low Latency Karatsuba Ofman Multiplier Variant for Elliptical Curve Cryptography", in Asian Journal of Computer and Information Systems (ISSN: 2321 – 5658)Volume 03– Issue 02, April 2015.

[13] Shahram Jahani, Azman Samsudin, and Kumbakonam Govindarajan Subramanian, "Efficient Big Integer Multiplication and Squaring Algorithms for Cryptographic Applications", in Hindawi Publishing Corporation Journal of Applied Mathematics Volume 2014, 24 July 2014.

[14] Shri Prakash Dwivedi1, "An Efficient Multiplication Algorithm Using Nikhilam Method", Fifth International Conference on Advances in Recent Technologies in Communication and Computing (ARTCom), 20-21 Sept, Bangalore, IET, ISBN: 978-1-84919-842-4, 223-228, July 2013.

[15] Gary C.T. Chow∗, Ken Eguro†, Wayne Luk∗and Philip Leong‡, "A Karatsuba-Based Montgomery Multiplier", in International Conference on Field Programmable Logic and Applications 2010.

[16] Sameh M. Shohdy, Ashraf B. El-Sisi, and Nabil Ismail, "Hardware Implementation of Efficient Modified Karatsuba Multiplier Used in Elliptic Curves", in International Journal of Network Security, Nov. 2010.

[17] Gutmann P., Cryptographic Security Architecture: Design and Verifcation, Springer-Verlag, 2004.

[18] Nadia Nedjah, Luiza de Macedo Mourelle, "A Review of Modular Multiplication Methods and Respective Hardware Implementations", April 18, 2005.

[19] Nedjah, N. and Mourelle, L.M. (Eds.), Embedded Cryptographic Hardware: Design and Security, Nova Science Publishers, Hauppauge, NY, USA, 2005.

[20] Nedjah, N. and Mourelle, L.M. (Eds.), New Trends on Embedded Cryptographic Hardware, Nova Science Publishers, Hauppauge, NY, USA (to appear).

[21] Dhem, J.F., Design of an efficient public-key cryptographic library for RISC-based smart cards, Ph.D. Thesis, Faculty of Applied Science, Catholic University of Louvain, May 1998.

[22] Menezes, A. van Oorschot, P. and Vanstone, S., Handbook of Applied Cryptography, CRC Press, 1996.

[23] Bewick, G.W., Fast multiplication algorithms and implementation, Ph. D. Thesis, Department of Electrical Engineering, Stanford University, United States of America, 1994.

[24] Barrett, P., Implementating the Rivest, Shamir and Aldham public-key encryption algorithm on standard digital signal processor, Proceedings of CRYPTO'86, Lecture Notes in Computer Science 263:311-323, Springer-Verlag, 1986.

[25] ElGamal, T., A public-key cryptosystems and signature scheme based on discrete logarithms, IEEE Transactions on Information Theory, 31(4):469-472, 1985.

[26] Knuth, D.E., The art of computer programming: seminumerical algorithms, vol 2, 2nd Edition, Addison-Wesley, Reading, Mass., 1981.

[27] Rivest, R., Shamir, A. and Adleman, L., A method for obtaining digital signature and public-key cryptosystems, Communications of the ACM, 21:120-126, 1978.

[28] W. Diffie and M.E. Hellman, New directions in cryptography, IEEE Transactions on Information Theory, vol. 22, pp. 644-654, 1976.

[29] Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (1974). The design and analysis of computer algorithms, Addison-Wesley.