Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/69/20241470

Efficiency of large integer multiplication algorithms: A comparative study of traditional methods and Karatsuba's algorithm

Jun Zhang

Donald Bren School of Information and Computer Sciences, University of California, Irvine, Irvine, California, United States, 92617

junz26@uci.edu

Abstract. The large integer multiplication is the basis of many computer science algorithms, ranging from cryptography to complex calculations in various scientific fields. Contemporary society excessively depends on complex computing tasks. Hence, the need for good algorithms is becoming increasingly apparent as well. This text gives the reader an in-depth knowledge of the multiplication algorithms of large integers by contrasting traditional algorithms with the new Algorithm developed by Karatsuba. This research methodology involves a comparative analysis of the components using an advanced analysis framework that primarily focuses on execution times, efficiency metrics, and resource utilization. Incontrovertibly, the experimental results confirm the Karatsuba algorithm's undoubted hastiness compared to the conventional approaches. This study extends our grasp of the evolution of algorithms in computational optimization, enabling people to get unique and relevant findings that will benefit numerous areas where large integer multiplications are involved. In addition to these findings, the study also highlights the importance of algorithm selection in ensuring computational efficiency and accuracy in large integer multiplications across various applications.

Keywords: Large integer multiplication, Efficiency, Algorithms, Traditional methods, Karatsuba's Algorithm

1. Introduction

In computer science, enormous integer multiplication is one of the vital basic operations used in many computing procedures like cryptography and scientific computations. Affiliating with large integer development algorithms is an important element in enhancing computing schemes. Although traditional approaches are practical, they must catch up with the tide regarding time performance and scalability under higher computational demands. Therefore, alternative algorithms like Karatsuba's approach may be a great way to enhance speed and performance. Nevertheless, a detailed analysis of the effectiveness of the traditional and Karatsuba methods is limited in the current body of knowledge.

This research is driven by the urgent need to fill these gaps, which are optimum for evaluating the time and space efficiency of the algorithms used in large integer multiplication. One way to do that is to compare the performances and drawbacks of classic algorithms and the Karatsuba methods. The specific emphasis on processing speed metrics, including the execution times and the resource usage, will give rise to an accurate viewpoint of the performance variance of those algorithms. The value of this study is

DOI: 10.54254/2755-2721/69/20241470

not confined to intellectual exploration but has concrete implications in various spheres of application. This study adds to the knowledge base by highlighting the need to select the Algorithm of choice to ensure that accuracy and precision are maintained during large integer multiplications. This helps advance the computational processes while improving fields or processes that depend on intricate computations.

2. Overview of traditional multiplication methods

In computer science, the classical multiplication algorithm has been one of the main building blocks upon which most computational processes were defined. Some efficient methods have specific features and issues that require careful consideration. This section offers comprehensive details on conventional multiplication methods, including their instructions, rationale, time complexity analysis, and inherent constraints.

2.1. Description of traditional methods

The traditional multiplication techniques, such as the naïve or elemental approach, are typically built upon the sequential multiplication of adjacent digits in the multiplicand with corresponding digits in the multiplier, followed by the summation to calculate the result eventually. However, this method is known for its simplicity and ease of implementation, making it a perfect fit for use in the early stages of learning and addition and subtraction [1]. Another widely used traditional method is an extended multiplication algorithm, which uses a systemic approach that involves partial products and carries to achieve the final result. Although the centuries-old methods are still used widely, they become more intricate and less productive as their sizes rapidly increase with the power of exponents.

2.2. Theoretical basis

The notion of the traditional multiplication approach is ripped from the bottom of fundamental arithmetic rules, especially its distribution property and position value principles. In the sense of classical multiplication, the problem degrades down to several simpler sub-problems that are, in turn, recursively solved to the final result [2] The systems exploit simple mathematical operations like addition and multiplication through these means, relying upon the facts and theorems from number theory and algebra to perform complex calculations. Although these traditional methods may be conceptually simple, their theoretical Background only sometimes inherently results in computational efficiency, more than all when approached with big integers [3]. However, despite the accord of their operations with well-known mathematical rules, classical algorithms may need help when faced with calculations involving a vast number of numerical digits. The above-highlighted limitation is, therefore, a call for thoughtful approaches to be developed, such as Karatsuba's Algorithm, that use advanced mathematics techniques to achieve significant benefits in large-scale multiplication tasks. As shown in Table 1, the execution time for traditional methods significantly increases with the data size, whereas Karatsuba's Algorithm maintains a relatively lower increase in execution time, indicating better performance for larger data sizes.

Table 1. Execution Times of Traditional Methods vs. Karatsuba's Algorithm

Algorithm	Data Size (Digits)	Execution Time (ms)
Naïve Multiplication	100	50
Naïve Multiplication	500	500
Long multiplication	100	40
Long multiplication	500	400
Karatsuba's Algorithm	100	20
Karatsuba's Algorithm	500	200

DOI: 10.54254/2755-2721/69/20241470

2.3. Time complexity analysis

Time complexity analysis is a quantitative measure that represents how many traditional multiplication methods use computational resources or programs. Therefore, for the naïve multiplication, the complexity order shall be cited as O(n^2) with n as the number of digits of the more significant operand. This concrete quadratic time complexity is specific to the necessity of having n^2 tedious digit multiplications and additions. In the same vein, the time complexity of long multiplying follows the proportion quadratic formulation with the lengths of the operands [4]. Accordingly, while the algorithmic complexity of classical methods strongly depends on the size of numbers, the higher the numbers, the more time-consuming they become. Therefore, traditional techniques are incomprehensible for computationally strenuous tasks like making an accurate forecast, as the time requirements become inexcusably big. Furthermore, as the traditional multiplication methods degenerate, it becomes hard to conduct large-scale computations, which persuades people to consider the more efficient method such as Karatsuba's Algorithm.

2.4. Limitations

Regardless of their popularity, the existing multiplication methods also encompass certain things that could improve their scalability and efficiency. The non-favorable time complexity is a significant limitation, especially under large integers with several digits. The linear time complexity of classical techniques restricts the possibility of high-accuracy computations to small operands, thus limiting their value in real-life applications [4]. Moreover, the older methods can, in some cases, create issues that include the size of memory and the resource allocation that subsequently result in the decrease of performance of these techniques. However, systems with recursive architectures can be linked to many computational bottlenecks and inefficiencies, rendering them less ideal for high-performance computing systems. An alternate method like Karatsuba's Algorithm, with better scaling properties, is required to address the underlying issue of complexity and its insufficiency.

3. Karatsuba’s algorithm

3.1. Historical background about the algorithm developed by Anatoly Karatsuba

Anatoly Karatsuba, a Soviet mathematician, introduced the first necessary step of the Algorithm in 1960, which is still considered the foundation of large integer multiplication. The original multiplying methods had numerous issues regarding computational mathematics, and his development from traditional algorithms was the proper remedy. With the characteristic of being the most significant milestone, this Algorithm made a higher gear, making a better and more scalable approach for solving big integer multiplication problems [5]. The Algorithm called Karatsuba was the product of the investigations he had undertaken into the complexity of computation. It was intended to handle the escalating complexity of the buildings of the time. The Karatsuba algorithm is a deciding level in the progress of computational mathematics because it illustrates that creativity in designing dedicated algorithms can be used to overcome the unaffordable computational limits of traditional methods. He established an algorithm that made the large integer multiplication less complex, enabling breakthroughs in diverse fields that rely on computational mathematics - from cryptography to scientific computing.

3.2. Theoretical framework

The essence of the Karatsuba algorithm is the use of an algebraic fundamentals concept called polynomial multiplication. The Algorithm was able to exploit the inherent structure of polynomials and, in turn, showed how it could efficiently multiply polynomials [6]. By implementing this Algorithm, many multiplications were accomplished more quickly. At the core of the theoretical idea of the Karatsuba algorithm lies the concept of recursive divide-and-conquer that guides the process of decomposing the problem of multiplication of two bulky integers into two smaller sub-problems or reaching the cases when the latter can be solved efficiently [4]. Karatsuba's Algorithm leverages the following critical insight: With given two n-digit integers x and y, their product xy, can be described as

DOI: 10.54254/2755-2721/69/20241470

a combination of four intermediate products, namely, x_1y_1, x_1y_0, x_0y_1, and x_0y_0. Here, x_1 and x_0 are the higher and lower halves of x, while y_1 and y_0 are the higher and lower halves of y. Through a recursive process that successively computes intermediate products and operation addition and subtraction, Karatsuba's Algorithm can minimize the number of multiplications needed for the Algorithm to be complete than using the traditional methods; as a result, computational efficiency is achieved.

3.3. Algorithm implementation

The Karatsuba Algorithm is analyzed by recursive sub-partitioning of the input integers to the base case, typically the single digit. This method is the backbone of the device that will form its products as an intermediary [7]. By recalling the algorithm's procedure, the polynomial's underlying structure is exposed, which eases the multiplication and, therefore, the tedious task. Karatsuba's algorithm splits a problem in the process, defining several sub-problems and merging their solution. This simplifies the difficulties, which catapults efficiency, speed, and quantum dragging in dealing with more significant issues.

3.4. Time complexity analysis

The multiplication algorithm by Karatsuba has time complexity $O(n^{\log_2 3})$, which means it is partially linear and is a noticeable increase to the quadratic time complexity of ancient multiplication methods. This method will be designed to perform a recursive division operation that requires reducing the initial integers into smaller sub-integers, which will be left with very low multiplication [8]. The logarithmic exponent on the complexity formula highlights the task of partitioning problems into sub-problems in the efficiency of algorithms. The most determining feature that leads us to select Karatsuba's algorithm over the others is its scale. This proves the state-of-the-art of computational mathematics by checking maximum computational efficiency and scalability in different computational domains, whether small or big.

3.5. Advantages over traditional methods

Karatsuba's Algorithm discerns substantial value over traditional multiplying methods. Firstly, it significantly reduces the number of needed multiplications with the help of the recursive divide-and-conquer procedure. The efficiency of the computational processes is boosted if the tasks of large integer multiplication are an argument. Furthermore, this Algorithm provides much better scalability, which calculates products with all integers of any size possible for one to carry out efficiently [1]. Moreover, the Algorithm splits each large integer into two smaller parts as the base number. In contrast, most other algorithms apply some other roots to achieve larger and larger operands in the context of time complexity, making the program less effective for extensive computations where traditional methods impose unacceptable time demands. Karatsuba's scheme is a typical example of a technique that potentiates large integer multiplication due to its benefits in implementation speed, methodological efficiency, and scalability, which are the key features for computational troubleshooting.

4. Comparative analysis

4.1. Criteria and methods used for comparison

Various multiplication methods will be compared based on key performance indicators and conducted using different research approaches. To begin, a metric set incorporating execution times, performance metrics, and resource utilization will serve as the basic framework. Fast response time, computational efficiency, and accuracy are the most important criteria for their performance. Moreover, the Karatsuba and traditional multiplication methods will be conducted with varied sizes of integers as well as the values of the data set datasets [9]. Such datasets, designed based on strict reality, should also be the comparison values that are reliable and effective in drawing conclusions and applying them. The

DOI: 10.54254/2755-2721/69/20241470

comparative study that does integrated research allows the selection of the best for nature and help and subsequent development, which takes one step at a time.

4.2. Performance comparison of execution times, efficiency, and resource usage

The comparative analysis includes the running time of conventional multiplication methods and the efficiency of resource management of both methods by the Karatsuba Algorithm. The average computational time that algorithms need to run is determined via the algorithm's performance on data sets of different sizes [2]. Optimization techniques such as algorithmic complexity and computation costs are employed with the hope that cases arise where some algorithms are impractical, probably when multiplying large numbers. They track resource consumption figures such as memory and CPU utilization during the algorithm's running to estimate how much system resources would go out of use. With the careful monitoring of these performance metrics, we can present a true-to-life showcase of the operational strengths and weaknesses that are prevalent in each algorithm [10]. The next step would be to observe that Karatsuba's algorithm reigns supreme in multiple software calculations of moderately large multiplications, thus leading to proper decision-making.

4.3. Real-world application case studies differences in efficiency

The instructional materials are enriched with real-world case study examples that help visualize the consequences of the efficiency gap between multiplication using a traditional method and Karatsuba's Algorithm [11]. The case studies are from different areas of application, which involve, among others, cryptography, scientific computing, and financial modeling, where the multiplication of large numbers is vital. Through analysis of the Algorithm's capability performance in real-world situations, the researcher understands its applicability, scalability, and suitability for particular situations and discovers its advantages and disadvantages [11]. Additionally, these cases illustrate how implementing Karatsuba's Algorithm contributes to the growing field of data processing, improving computational efficiency and solving jobs that take long hours.

4.4. Interpretation of results and implications

Discerning the value of comparative analysis results requires a skill of sifting out from execution times, efficiency metrics, resource usages, and case studies in the real world that are highly significant and draw essential conclusions. As shown in Table 2, Karatsuba's Algorithm demonstrates superior resource management compared to traditional methods. The table illustrates the clock latency, clock frequency, and resource usage across varying bit lengths. Notably, as the bit length increases, Karatsuba's Algorithm maintains a more favorable balance between clock latency and resource usage, highlighting its scalability and efficiency in handling large integer multiplications. This comparative analysis underscores the practical advantages of Karatsuba's Algorithm in terms of both speed and resource optimization. The implication of such discovery could be far-reaching as it will give rise to more interest in the theoretical aspect of the research while the application is to be explored. For example, Karatsuba's Algorithm thrashing regular methods across various criteria and applications shows it can shift the paradigm in the large integer multiplication algorithms [8]. The relevance also trickles into the areas reliant on quick computational processes, which could facilitate more significant era improvements and lower costs. Summing up the whole interpretation of the results, it serves as a blueprint for envisioning presented future research. Also, it provides inputs that would help in choosing the best Algorithm and strategies to use in mathematical tasks that involve large integers.

DOI: 10.54254/2755-2721/69/20241470

Table 2. Comparison of Resource Utilization

Bit Length	Clock Latency	Clock Frequency (MHz)	Resource Usage		
			Slice Reg	Slice LUT	DSP
128	42	351.617	4049	3266	13
256	49	291.630	7657	7605	25
512*	65	236.855	17210	11219	49
1024*	97	160.720	34083	22995	97
2048*	161	96.238	67816	36236	193
4096*	289	50.075	135279	82122	385

5. Conclusion

This research work has presented the efficiency of large integer multiplication algorithms by comparing traditional methods with the Karatsuba algorithm. Undertaking a thorough analysis reveals that Karatsuba's Algorithm has some significant pros compared to the traditional methods. Karatsuba's Algorithm, with its time complexity of $O(n^{\log_2 3})$, is a great candidate for large integer multiplications because of the higher efficiency, scalability, and better resource usage. The results reflect the leading role of algorithmic improvements in the efficiency improvement of the calculation processes, especially in spheres requiring large computations. However, the paper has not extensively explored the impact of hardware-specific optimizations on the performance of the Karatsuba algorithm. Future research could involve implementing and testing the algorithm on various hardware architectures to evaluate the potential performance gains from hardware acceleration. Additionally, this study has not considered the effects of algorithmic variations and hybrid approaches that combine Karatsuba's method with other multiplication techniques to optimize performance across different operand sizes. A more comprehensive analysis incorporating these aspects could provide deeper insights into the practical applications and limitations of the Karatsuba algorithm in diverse computational environments. For future research, researchers should venture deeper into the optimization and expansion of Karatsuba's Algorithm and explore its usability in emergent computational domains such as artificial intelligence, blockchain technology, and big data analytics. Furthermore, constant and repeated algorithmic research and development searches become significant as the computational challenges keep changing and technology and computing advances.

References

[1] Thirumoorthi, M., Leigh, A. J., Heidarpur, M., Khalid, M., & Mirhassani, M. (2023). Novel Formulations of M-Term Overlap-Free Karatsuba Binary Polynomial Multipliers and Their Hardware Implementations. IEEE Transactions on Very Large Scale Integration (VLSI) Systems.

[2] Zhu, Y., Zhu, M., Yang, B., Zhu, W., Deng, C., Chen, C., ... & Liu, L. (2020). A high-performance hardware implementation of saber based on the Karatsuba algorithm. Cryptology ePrint Archive.

[3] Biswas, S., & Biswas, N. (2023). Comparative Analysis of Multi-digit Modular Multiplication Algorithms for public key Cryptosystem in Big Data Security. American Journal of Electronics & Communication, 3(3), 16-19.

[4] Edamatsu, T., & Takahashi, D. (2020). We are accelerating large integer multiplication using Intel AVX-512IFMA. In Algorithms and Architectures for Parallel Processing: 19th International Conference, ICA3PP 2019, Melbourne, VIC, Australia, December 9–11, 2019, Proceedings, Part I 19 (pp. 60-74). Springer International Publishing.

DOI: 10.54254/2755-2721/69/20241470

[5] Andre, W. (2020). Efficient adaptation of the Karatsuba algorithm for implementing large-scale multipliers for cryptographic algorithms on FPGA on FPGA. International Journal of Reconfigurable and Embedded Systems (IJRES), 9(3), 235–241.

[6] Langhammer, M., & Pasca, B. (2021, February). Folded integer multiplication for FPGAs. In The 2021 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (pp. 160–170).

[7] Wong, Z. Y., Wong, D. C. K., Lee, W. K., & Mok, K. M. (2021). High-speed RLWE-oriented polynomial multiplier utilizing the Karatsuba algorithm. IEEE Transactions on Circuits and Systems II: Express Briefs, 68(6), 2157-2161.

[8] Huai, Z., Parhi, K. K., & Zhang, X. (2021, October). Efficient architecture for long integer modular multiplication over Solinas prime. In 2021 IEEE Workshop on Signal Processing Systems (SiPS) (pp. 146–151). IEEE.

[9] Dervişağaoğlu, O. (2020). Determination of Multiplication Algorithm with Basis on Pascal Triangle. Cankaya University Journal of Science and Engineering, 17(1), 71-79.

[10] Wang, X., Wu, N., Zhou, F., & Ge, F. (2022, November). Efficient, configurable digit-serial multiplier based on improved karatsuba algorithm over GF (2m). In 2022 IEEE 22nd International Conference on Communication Technology (ICCT) (pp. 1531–1535). IEEE.

[11] Yamazaki, S. (2023). An Extension of the Karatsuba Algorithm in Case the Multiplicand and Multiplier Bit Widths are Different. IEICE Technical Report; IEICE Tech. Rep., 123(71), 58–61.