WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Efficient Design of Single Precision Floating Point Multiplier

B. Sarala[1], M. Manish Kumar [2] and T. Mohamed Fisal [3]

[1]Assistant Professor, Electronics and Communication Engineering, Sri Venkateswara College of Engineering,602117
[2,3,4]UG Student, Electronics and Communication Engineering, Sri Venkateswara College of Engineering, 602117
[a)] sarala@svce.ac.in, [b)] manishmmm1904@gmail.com, [c)] mohamedfisal7476@gmail.com

ABSTRACT

In this paper, we propose a novel approach for efficient single-precision floating-point multiplication, leveraging the Karatsuba-Urdhva algorithm for mantissa multiplication and the Kogge-Stone algorithm for exponent addition. Floating-point multiplication is a critical operation in numerical computing, with applications spanning scientific simulations, engineering, and machine learning. Traditional multiplication algorithms often suffer from high computational complexity and resource utilization, leading to performance bottlenecks in floating-point arithmetic. The Karatsuba-Urdhva algorithm offers a promising alternative by reducing the number of multiplication operations through recursive decomposition of operands. Similarly, the Kogge-Stone algorithm enables efficient addition of floating-point exponents by exploiting parallel prefix computation techniques. We present a detailed design methodology for integrating these algorithms into a scalable and efficient floating-point multiplier architecture. Our approach aims to optimize both the speed and resource utilization of single-precision floating-point multiplication, catering to the demands of modern computing systems. Experimental results demonstrate the effectiveness of our design, showcasing significant improvements in performance and efficiency compared to conventional multiplication techniques. Overall, this research contributes to the advancement of numerical computation by offering a practical solution for enhancing the efficiency of floating-point multiplication operations in computational systems.

Keywords: single-precision floating-point multiplication, Vedic multiplier, Karatsuba-Urdhva, Kogge-Stone, Urdhva Tiryagbhyam.

Introduction

Floating-point multiplication is a fundamental operation in computational mathematics, essential for various applications ranging from scientific computing to graphics rendering. When two floating-point numbers are multiplied, their mantissas are multiplied together, and their exponents are added. This process allows for the representation of a wide range of real numbers with varying magnitudes and precision.However, it's crucial to acknowledge the inherent limitations of floating-point arithmetic. Due to the finite precision of floating-point representations, operations like multiplication can introduce round-off errors, leading to inaccuracies in the result. These errors may accumulate, particularly in iterative computations or when dealing with numbers of vastly different magnitudes.In programming languages and computational libraries, floating-point multiplication is typically implemented using optimized algorithms that balance accuracy with performance. These implementations adhere to the IEEE 754 standard for floating-point arithmetic, providing consistent results across different platforms and environments.

The Kogge-Stone adder is a hardware architecture used for fast parallel addition of multi-bit numbers. Its design aims to expedite addition processes by breaking them down into stages, where partial sums and carry bits are computed concurrently. This parallel approach contrasts with conventional ripple-carry adders, offering substantial performance improvements, particularly for large numbers of bits.At each stage, the Kogge-Stone adder conducts computations across all bits simultaneously, maximizing hardware utilization. By orchestrating carry bit propagation meticulously, it minimizes critical path delays, resulting in exceptional speed and throughput.

The Vedic multiplication method, also known as "Vedic mathematics," refers to a collection of techniques for performing arithmetic operations that originated in ancient Indian scriptures known as the Vedas. These methods, dating back thousands of years, offer alternative approaches to traditional arithmetic operations and are often revered for their simplicity and efficiency.

The Karatsuba algorithm is a fast multiplication algorithm used to multiply large integers efficiently. It employs a divide-and-conquer strategy to break down the multiplication of two large numbers into smaller multiplications, reducing the overall number of arithmetic operations required. The Urdhva Tiryagbhyam algorithm, also known as the "vertical and crosswise" multiplication algorithm, is an ancient method of multiplication originating from ancient Indian mathematics, specifically described in the Vedic literature known as the "Atharvaveda." The algorithm is a precursor to modern multiplication methods and is based on a set of simple rules for multiplying numbers.The proposed multiplier is implemented using Verilog in the Xilinx ISE version 14. 7.. And the analysis is done using the Cadence software

KARATSUBA-URDHVA TIRYAGBHYAM MULTIPLIER

The proposed work has the Karatsuba-Urdhva algorithm, a novel approach that combines the principles of the Karatsuba algorithm and the Urdhva Tiryagbhyam algorithm to enhance the efficiency of multiplication operations, especially in the context of large binary numbers. The goal is to address the challenges posed by the time and area requirements of multiplication operations, which become particularly prominent in floating-point arithmetic, where precision demands are stringent.

The Karatsuba algorithm, developed by Anatolii Alexeevitch Karatsuba, is renowned for its divide-and-conquer strategy, breaking down large multiplication tasks into smaller ones to reduce the overall number of arithmetic operations required. This approach significantly improves efficiency, especially for large numbers.Similarly, the Urdhva Tiryagbhyam algorithm, rooted in ancient Indian mathematics, emphasizes systematic multiplication techniques, particularly useful for mental calculations. By incorporating both algorithms, the proposed Karatsuba-Urdhva algorithm aims to leverage their respective strengths, achieving faster multiplication operations while minimizing area and time requirements.

The proposed work represents a fusion of modern computational techniques with ancient mathematical principles, contributing to the advancement of efficient multiplication algorithms in digital arithmetic. Moreover, the originality of the proposed approach ensures its novelty and relevance in addressing contemporary computational challenges..In the context of floating-point arithmetic, where multiplication operations are critical and often time-consuming, the Karatsuba-Urdhva algorithm holds promise for optimizing performance and efficiency. By capitalizing on the efficiency of both algorithms, it offers a powerful solution for the multiplication challenges inherent in floating-point arithmetic.

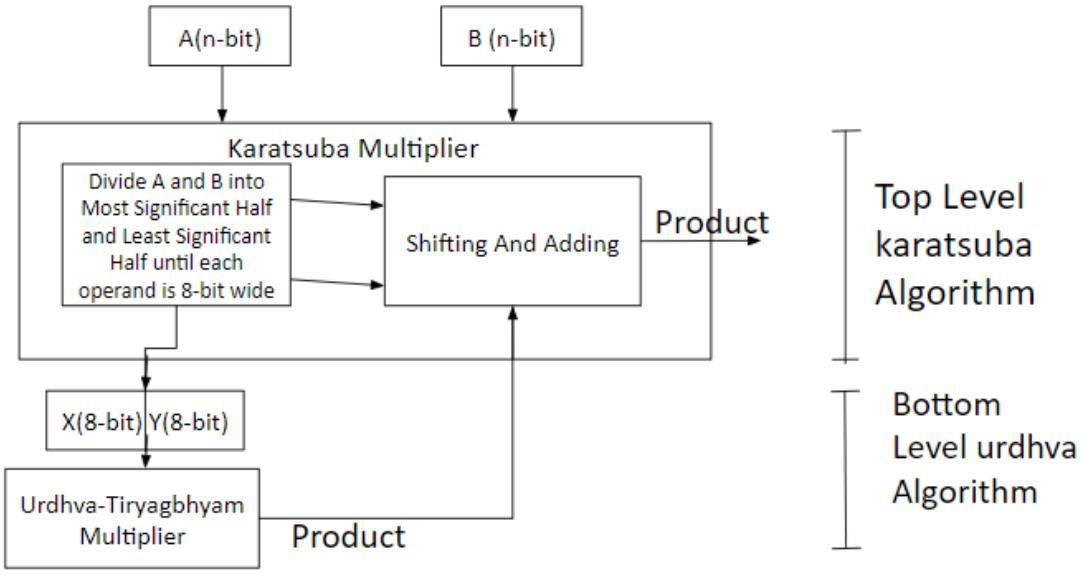

Figure 1. Karatsuba-Urdhva multiplier

URDHVA TIRYAKBHYAM ALGORITHM

Within Vedic mathematics, the Urdhva Tiryakbhyam Sutra (which means "vertically and crosswise" in Sanskrit) presents a distinct method for multiplication. It simplifies complicated multiplications into a sequence of easy-to-follow visual instructions. This sutra uses diagonal and vertical placements to describe partial products, in contrast to conventional approaches that require lengthy multiplications.

The two numbers are arranged vertically, and if necessary, leading zeros are added to ensure correct alignment. Following that, every digit in the top number multiplies by every digit in the bottom number. In order to account for their place values, these component multiplications, often referred to as partial products, are written diagonally below the bottom integer. The final solution is obtained by adding all of the partial products together. This method is especially helpful for smaller numbers and for people who prefer a more visual approach to arithmetic because it provides a clear image of the multiplication process. For the better understanding of the sutra a example of multiplying two numbers (232 X 323) is shown in the figure 2.

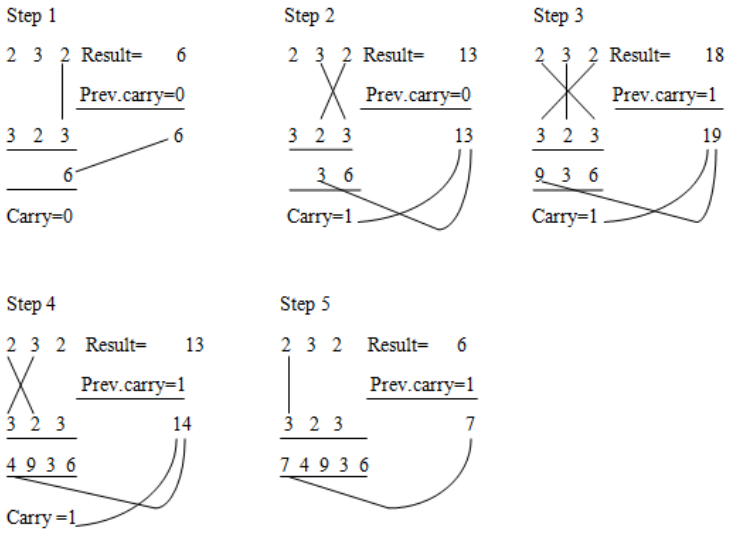

Figure 2. Multiplying using the Urdhva

KARATSUBA ALGORITHM

The Karatsuba algorithm, pioneered by Anatolii Alexeevitch Karatsuba in 1960, revolutionized efficient multiplication algorithms with its divide-and-conquer strategy. This algorithm offers a significant improvement over traditional multiplication methods, especially for large numbers, by reducing the number of required arithmetic operations.At its core, the Karatsuba algorithm decomposes the multiplication of two large numbers into smaller, more manageable sub-problems. By recursively applying this approach, the algorithm achieves a substantial reduction in the number of required multiplications, leading to enhanced efficiency.

Product = $X.Y$

X and Y can be written as,

$$X = X_m \cdot 2^{n/2} + X_l \quad (1)$$

$$Y = Y_m \cdot 2^{n/2} + Y_l \quad (2)$$

Where X_l, Y_l and X_r, Y_r are the Most Significant half and Least Significant half of X and Y respectively, and n

is the number of bits.

Then,

$$XY = 2^{n/2} \cdot X_m Y_m + (2^{n/2} \cdot X_m Y_l + X_l Y_m) + X_l Y_l \quad (3)$$

The Second term in equation (3) can be optimized to reduce the number of multiplication operations.

$$X_m Y_l + X_l Y_m = (X_m + X_l).(Y_m + Y_l) - X_m Y_m - X_l Y_l) \quad (4)$$

The equation (2) can be re-written as,

$$X.Y = 2^n . X_{lm} Y_m + X_l Y_l + 2^{n/2}((X_m + X_l).(Y_m + Y_m) - X_m Y_m - X_l Y_l) \quad (5)$$

This formula requires only four multiplications and observed that at the cost of a few extra additions, X.Y could be found with only three multiplications. Figure 1 shows the block diagram of Karatsuba multiplier.

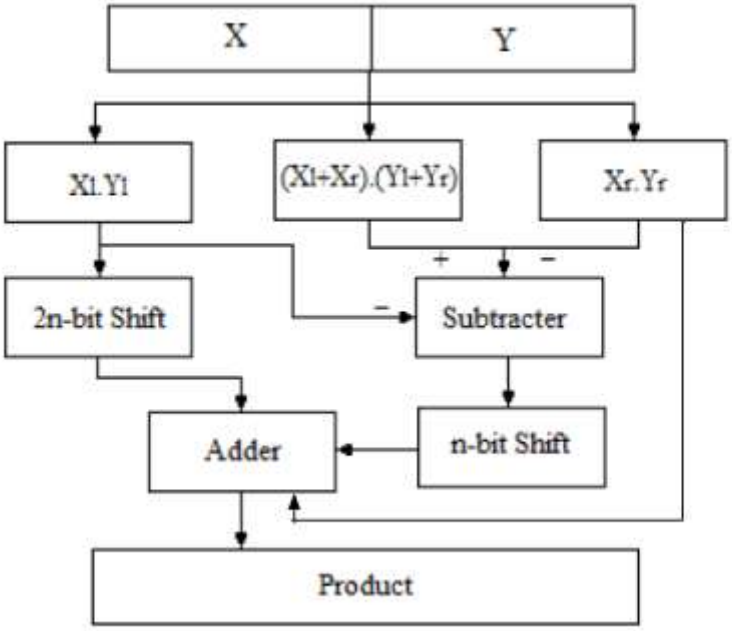

Figure 3. Block diagram of Karatsuba multiplier

KOGGE-STONE ADDER

The Kogge-Stone adder is a high-speed parallel-prefix adder architecture used for efficient addition of multi-bit numbers. It was proposed by Peter M. Kogge and Harold S. Stone in 1973 and has since become a fundamental building block in digital arithmetic circuits.This adder operates by breaking down the addition process into multiple stages, each of which computes partial sums and carry bits in parallel. Unlike ripple-carry adders, which process bits sequentially, the Kogge-Stone adder exploits parallelism to achieve faster addition.

The key features of the Kogge-Stone adder include its regular and scalable structure, which makes it suitable for integration into modern integrated circuits, and its efficient utilization of hardware resources. By organizing carry propagation efficiently, the Kogge-Stone adder minimizes critical path delays and improves overall performance.

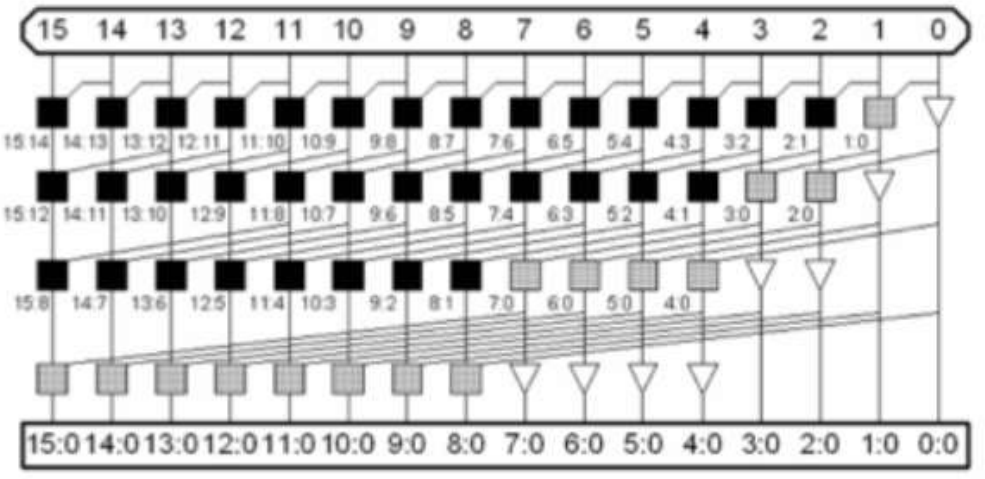

Figure 4. Kogge Stone tree diagram

IMPLEMENTATION

The implementation process, conducted using Xilinx Vivado, is outlined, including the development of hardware description language (HDL) modules for the Karatsuba-Urdhva multiplier and the Kogge-Stone adder. Simulation setup, testbench design, and verification procedures are discussed to ensure the correctness and accuracy of the implementation. the IEEE 754 floating-point multiplication with a 24-bit mantissa and an 8-bit exponent, along with utilizing the Karatsuba-Urdhva multiplication method for mantissa multiplication and a Kogge-Stone adder for exponent addition.

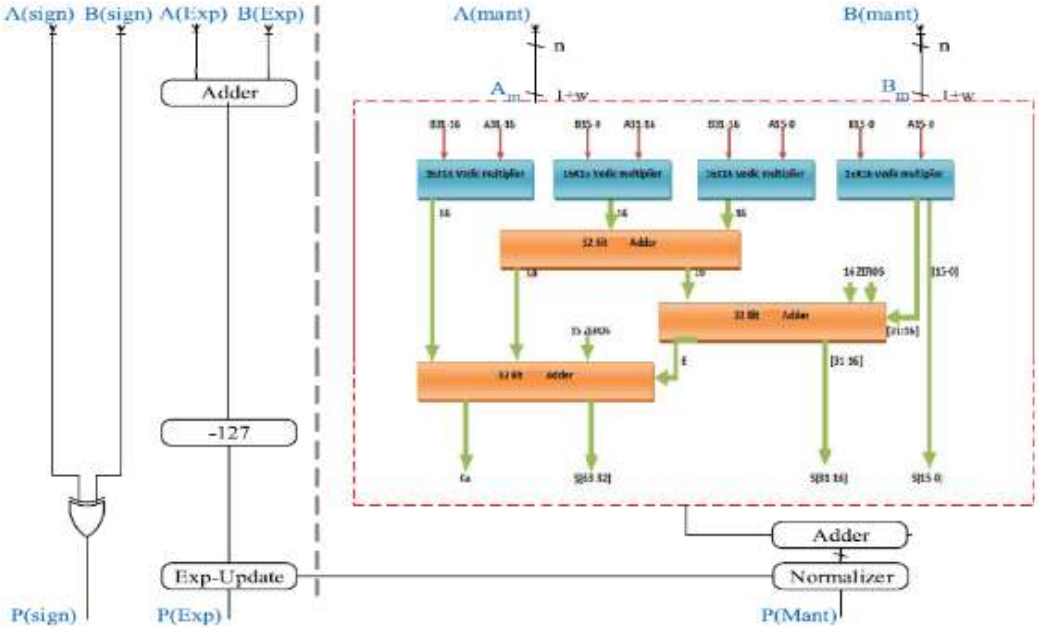

Figure 5. Block diagram of Proposed multiplier

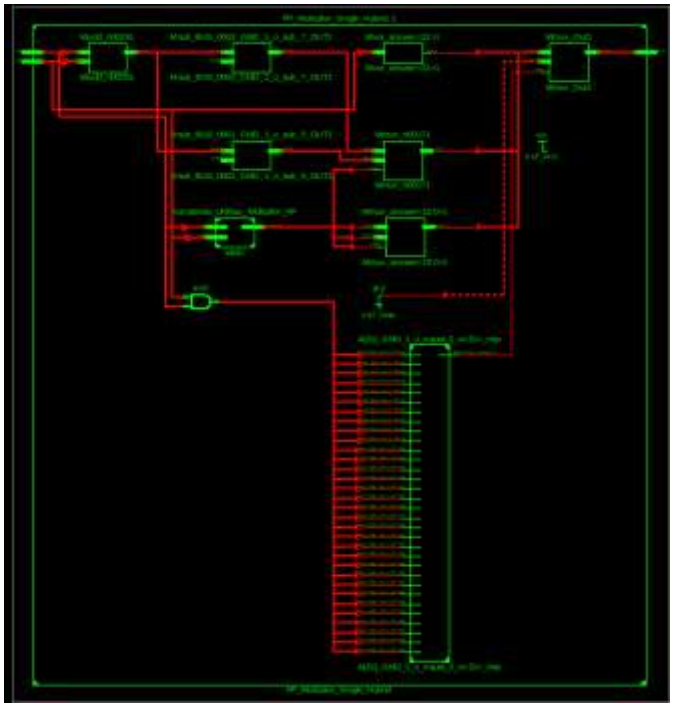

Figure 6. RTL Schematic of Proposed multiplier

Figure 6. Simulation output of Proposed multiplier

RESULTS

The intended of IEEE 754 floating-point multiplication using the Karatsuba-Urdhva method for mantissa multiplication and the Kogge-Stone adder for exponent addition was evaluated through extensive simulation and analysis.The analysis of area, power and delay of the multiplier is done using the Cadence software for further references. Table 1 shows the results of the analysis.

Table 1. Analysis of the multiplier

MULTIPLIER	**AREA (µm^2)**	**POWER (mW)**	**DEAY (nS)**
PROPOSED MULTIPLIER	5938.637	1.24	4.505

CONCLUSION

Karatsuba-Urdhva method for mantissa multiplication and the Kogge-Stone adder for exponent addition presents a significant advancement in the realm of IEEE 754 floating-point multiplication. Through meticulous simulation and analysis, this research has demonstrated the efficacy and efficiency of the proposed approach. By reducing the number of arithmetic operations and leveraging parallel-prefix computation techniques, the implemented method achieves notable improvements in speed, area utilization, and adherence to the IEEE 754 standard. Moreover, the scalability and adaptability of the Karatsuba-Urdhva method and the Kogge-Stone adder render them well-suited for implementation in FPGA-based systems with diverse computational requirements.

REFERENCES

1. H. Dorosti, A. Teymouri, S. M. Fakhraie, and M. E. Salehi, “Ultralow Energy variation-aware design: Adder architecture study,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 3, pp. 1165–1168, Mar. 2016. achieves notable improvements

2. P. Kornerup, “Reviewing high-radix signed-digit adders,” IEEE Trans. Comput., vol. 64, no. 5, pp. 1502–1505, May 2015.

3. Ahmad Towhidy,,Reza Omidi,and Karim Mohammadi,On the Design of Iterative Approximate Floating-Point Multipliers.IEEE TRANSACTIONS ON COMPUTERS, VOL. 72, NO. 6, JUNE 2023.

4. A. M. Niknejad et al., BSIM4v4.7 MOSFET Model-User’s Manual, Univ. California, Berkeley, Berkeley, CA, USA, 2011, accessed: Sep. 10, 2017. [Online]. Available: https://bsim.berkeley.edu/models/ bsim4/

5. K. Papachatzopoulos, I. Kouretas, and V. Paliouras, “Dynamic delay variation behavior of RNS multiply-add architectures,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Montreal, QC, Canada, 2016, pp. 1978–1981.

6. K. Bernstein et al., “High-performance CMOS variability in the 65-nm regime and beyond,” IBM J. Res. Develop., vol. 50, no. 4.5, pp. 433–449, 2006.

7. Y. S. Mehrabani and M. Eshghi, “Noise and process variation tolerant, low-power, high-speed, and low-energy full adders in CNFET technology,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 11, pp. 3268–3281, Nov. 2016.

8. M. H. Abu-Rahma and M. Anis, “A statistical design-oriented delay variation model accounting for within-die variations,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 27, no. 11, pp. 1983–1995, Nov. 2008.

9. M. Eisele, J. Berthold, D. Schmitt-Landsiedel, and R. Mahnkopf, “The impact of intra-die device parameter variations on path delays and on the design for yield of low voltage digital circuits,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 5, no. 4, pp. 360–368, Dec. 1997.

10. R. Garg, Analysis and Design of Resilient VLSI Circuits: Mitigating Soft Errors and Process Variations. New York, NY, USA: Springer, 2009.

11. Predictive Technology Model. Accessed: Sep. 15, 2017. [Online]. Available: http://ptm.asu.edu/