IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13613

Edge Case Optimization in 8-bit Multipliers Leveraging Vedic Mathematics

Nishanth Rao[1], S. G. Raghavendra Prasad[2]

Student, Information Science and Engineering, R.V College of Engineering, Bengaluru, India[1]

Assistant Professor, Information Science and Engineering, R.V College of Engineering, Bengaluru, India [2]

Abstract: Vedic Mathematics is an ancient Indian mathematical system known for its unique and efficient methods of computation. This paper presents the design of an 8-bit multiplier that makes use of the Nikhilam Sutra (NS) to create a fast constant coefficient (with 0xFF) and the Yavadunam Sutra (YS) to create a fast squarer which accepts inputs between 0xF0 and 0xFF. The top module decides which module to use based on the input values and makes use of IP Core for standard multiplication. The designs are implemented in Verilog and implemented in Xilinx Vivado, making use of the Atrix-7 FPGA board. The results show improvement in the NS and YS module, in terms of utilization, power consumption, and path delay. The top module, however, suffers from overheads due to multiplexing and comparisons, which offset the performance gains observed.

Keywords: Vedic, Multipliers, Yavadunam, Nikhilam, Verilog, DSP

I. INTRODUCTION

Vedic Mathematics consists of 16 “sutras” or methods for finding solutions to mathematical calculations, known for their uniqueness and efficiency. Among its sutras, two stand out for their application in multiplication: Nikhilam and Yavadunam.

The Nikhilam sutra (henceforth, NS) translates to "all from 9 and the last from 10" and is effective for multiplying numbers close to a base, such as 10, 100, or 1000. It leverages the difference between a given number and the base, allowing for quick and efficient calculations. The Yavadunam sutra (henceforth YS), meaning "whatever the extent of its deficiency," deals with numbers that have a deficiency from a base. These sutras offer pattern-based methods for multiplication that can be advantageous in digital arithmetic. These methods can be extended to base 16. This is especially useful since this allows for applying the methods on a per nibble basis.

Over the years, there have been several controversies regarding the historical validity of vedic mathematics [12]. The authors do not claim to assert any statements with respect to the historical origins of vedic mathematics. What is of concern here is only the number theoretical aspects.

The purpose of this study is to explore and evaluate the efficiency of Vedic Mathematics-based multiplication methods, specifically the Nikhilam and Yavadunam sutras, in comparison with standard multiplication technique in digital systems. By developing a simulation model of Vedic Mathematical multiplication, the study aims to conduct a comparative analysis of performance, including speed, accuracy, and resource utilization.

II. LITERATURE SURVEY

In [1], the focus is on the significance of low-power, high-frequency multipliers in VLSI design. The study analyzes and compares Wallace Tree, Array, and Baugh-Wooley multipliers to assess their suitability and efficiency. Physical verification of the designs is conducted to determine performance in terms of low power, area, and speed using simulations in ModelSim and Xilinx ISE. Results reveal that the Baugh-Wooley multiplier has the lowest power consumption and delay, while the Array multiplier is most efficient in minimizing area. Each multiplier offers distinct strengths and weaknesses for different applications, suggesting that future research could optimize these designs.

In [2], the focus is on the role of multipliers in digital signal processors (DSPs) and the importance of high-speed multipliers to enhance DSP performance. To improve digital circuits' speed while reducing on-chip area and memory consumption, the paper explores Vedic mathematical algorithms, particularly "Urdhva Tiryakbhyam" and "Nikhilam Navatashcaramam Dashatah." These algorithms are implemented using Verilog language and tested with Xilinx ISE 14.5.

 This work is licensed under a Creative Commons Attribution 4.0 International License

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024
DOI: 10.17148/IJARCCE.2024.13613

The study compares the two algorithms' performance in terms of propagation delay, revealing that "Urdhva Tiryakbhyam" is more efficient for smaller inputs, while "Nikhilam Navatashcaramam Dashatah" excels with larger inputs. The results demonstrate that Urdhva Tiryakbhyam is superior in terms of area and speed for 8-bit multipliers compared to conventional multipliers, while Nikhilam offers better combinational path delay and lower area utilization. Future work includes extending this research to create a 64-bit ALU and an integrated multiplier based on the two Vedic sutras.

In [3], the increasing significance of digital electronics in daily life is highlighted, along with the critical role of digital multipliers in modern devices, digital signal processing, and other applications. The paper reviews and compares the performance of various multipliers, focusing on key metrics such as delay, power consumption, and required area. This comparative analysis brings together important digital multipliers to identify the most suitable option for specific applications, aiding in the selection process.

The authors of [4] focus on enhancing high-speed applications by implementing digital multipliers based on Vedic mathematics, specifically the Yavadunam algorithm for squaring binary numbers. Despite the widespread use of complex multiplications, there has been a lack of hardware implementation for this squaring algorithm. This paper addresses this gap by designing a squaring architecture using Yavadunam, which is simulated and realized with the Xilinx Spartan 3E kit. The study compares the existing and proposed Yavadunam architectures through simulated outputs and synthesis reports.

In [5], the research explores the use of Vedic mathematical sutras to design and execute multiplications and divisions, thereby improving VLSI design circuits. The paper surveys several implementations of Vedic mathematics in digital circuits, analyzing architectural aspects such as area, power, and speed.

In [6], the authors introduce a new approach to the squaring technique in Vedic mathematics, specifically focusing on the Dwandwa Yoga sutra. This technique is applied to the squaring of two-digit numbers, utilizing the Xilinx Integrated Synthesis Environment design suite (14.7) for design and simulation. The proposed Vedic multiplier architecture shows improvements in speed and requires fewer components compared to conventional multipliers. The time taken for the squaring operation is reduced, demonstrating the efficacy of employing Vedic algorithms.

In [7], a Vedic multiplication algorithm using the Urdhava Tiryakbhyam method is proposed to improve the speed of Digital Signal Processors (DSPs). This method, rooted in Vedic mathematics, enhances binary number multiplication in digital hardware systems by reducing combinational path delay compared to existing multipliers. By applying the ancient Indian approach of vertical and crosswise multiplication, the proposed algorithm decreases processing time and improves computation speed.

S Akhter et al[8], introduces a modified binary multiplier based on Vedic mathematics, aiming to improve the efficiency of previous Vedic multiplier circuits. The proposed modification focuses on enhancing speed and area efficiency. The implementation uses VHDL, with simulations carried out in the Mentor Graphics ModelSim tool and circuit synthesis done using the Xilinx ISE Design Suite 14.1. Simulations were conducted for 4-bit, 8-bit, and 16-bit multiplication operations, demonstrating improved speed and device utilization compared to previous Vedic multipliers. The paper provides simulation waveforms for the 4-bit operation and suggests that the method can be scaled to larger bit sizes.

The work [9] is considered to be the authoritative source of the Vedic mathematics sutras, consisting of examples of how to apply them.

Behrooz Parhami[10] exposits on digital number representation and standard digital arithmetic methods, including addition, subtraction and multiplication, which are relevant to this study.

III. DESIGN AND IMPLEMENTATION

All the designs are made in Verilog, and simulated and implemented in Xilinx Vivado 2023.2. Artix-7 is chosen to be the board on which the circuits are implemented due to its nature of being a common board.

A. NS Module

The NS module must implement Nikhilam Sutra for cases where one of the numbers is 0xFF. This simplifies the operations involved. First, 1 must be subtracted from the first number. This number acts as the upper byte. The lower byte is then calculated as the difference between the 0xFF and the newly calculated upper byte. The difference can be

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13613

calculated as the value obtained on XORing the two numbers. The upper and lower byte together form the 16-bit output. For example: 0x23 multiplied by 0xFF would yield 0x22 as the upper byte (by subtracting 1) and 0xDD as the lower byte. The output will thus be 0x22DD. Fig. 1 is the RTL schematic of the NS module.

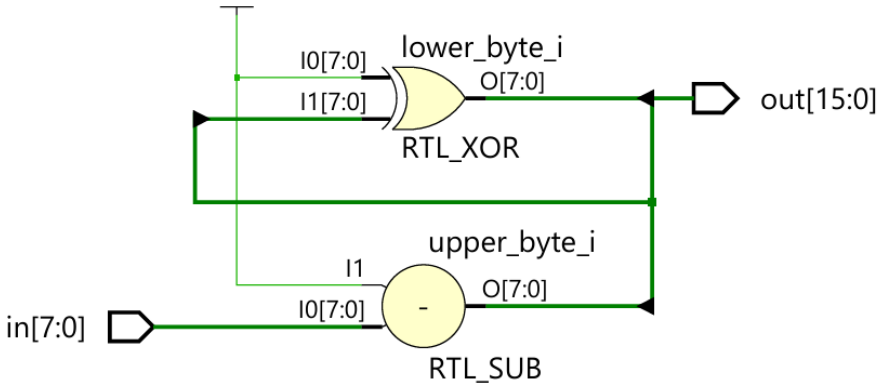


Fig. 1 NS Module RTL Schematic

B. YS Module

The deficiency is calculated as the difference between 0x10 and the lower nibble of the input. The upper byte is then calculated as the difference between input and deficiency. The lower nibble is calculated as the square of the 5-bit number using a hard-coded 0-0x10 fast multiplier. The lower byte may overflow, so the 9th bit is added to the upper byte. The output is the combination of upper byte and lower byte. For example: 0xFF is to be squared. The deficiency is 0x1. The upper byte is 0xFF - 0x1 which is 0xFE and the lower byte is deficiency squared which is 0x1. The final output will then be 0xFE01. Fig. 2 is the RTL schematic for the YS module.

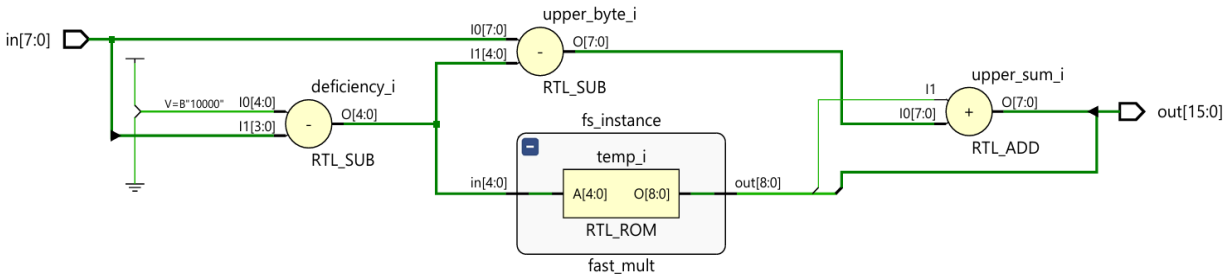


Fig. 2 YS Module RTL Schematic

C. Inbuilt Multiplier (IP Core)

The Inbuilt module is a pure combinational circuit (0 pipeline stages). This serves as a benchmark for comparative analysis.

D. Top Module

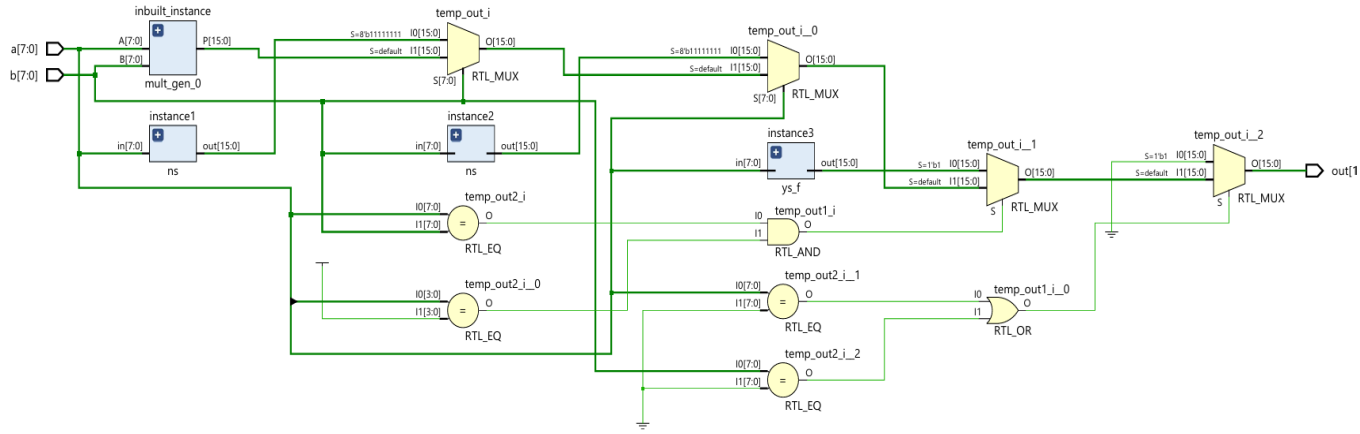
The top module is responsible for bringing together NS, YS and IP Core multiplier together. The top module decides which module to use based on the input values. The multiplexing adds overhead which may offset the performance gains.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13613



IV. RESULTS AND ANALYSIS

Table I summarizes the performance of the modules. The NS and YS modules utilize a significantly reduced number of LUTs. They use less power and have a smaller path delay compared to the standard multiplier. Path delay for the critical path is what determines the throughput in this case. The Max condition is the longest delay under worst case scenario, Min condition is the longest delay under best case scenario. The modules have the highest throughput (over their limited domain). The performance gains seen in the NS and YS module is offset in the top module due to the overheads accrued due to multiplexing and performing comparisons.

TABLE I COMPARITIVE PERFORMANCE SUMMARY

Performance Parameters	NS Module	YS Module	Inbuilt 8x8 Multiplier (Pure Combinational) (IP Core)	Top
LUTs Used	10	12	60	127
Power (W)	Signals: 0.110 Logic: 0.043 I/O: 5.068	Signals: 0.113 Logic: 0.062 I/O: 6.674	Signals: 0.530 Logic: 0.421 I/O: 13.224	Signals: 0.110 Logic: 0.043 I/O: 5.068
Path Delay (Critical Path) (ns)	Max: 7.470 Min: 2.396	Max: 7.821 Min: 2.531	Max: 10.922 Min: 2.855	Max: 12.533 Min: 3.355

V. CONCLUSION

The NS Module acts as a fast and efficient constant coefficient 8-bit multiplier. Constant coefficient multipliers find use in numerical algorithms to scale a variable with constant, especially in digital signally processing [11]. YS module acts as an efficient squarer for inputs in the range 0xF0 to 0xFF.

Apart from DSP, this specialized squarer can be effectively utilized in cryptographic algorithms, where modular arithmetic operations often involve squaring large numbers. The top module aims to use the modules in their input range. Due to the multiplexing and comparison operations that are required for the same, the advantages of the modules are offset. Despite this, the modules of their own serve as a great specific case multiplier.

The top module may be improved with better multiplexing and reduced comparison, making it more viable. The bus size may be increased to operate on higher bit numbers. This is still not a free-size scalable solution. A more accurate understanding of the performance may be obtained by implementing the design on different boards of various performance capabilities.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.102 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 6, June 2024

DOI: 10.17148/IJARCCE.2024.13613

REFERENCES

[1]. Rajasree Shanmuganathan A, Kathirvel Brindhadevi, "Comparative analysis of various types of multipliers for effective low power", Elsevier, Microelectronic Engineering Volume 214, 1 June 2019, Pages 28-37

[2]. N. R. Mistri, S. B. Somani and V. V. Shete, "Design and comparison of multiplier using vedic mathematics," 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 2016, pp. 1-5, doi: 10.1109/INVENTIVE.2016.7824870.

[3]. Vikas Kaushik, Himanshi Saini, "A Review on Comparative Performance Analysis of Different Digital Multipliers", Advances in Computational Sciences and Technology, ISSN 0973-6107 Volume 10, Number 5 (2017) pp. 1257-1272

[4]. A. Deepa and C. N. Marimuthu, "VLSI Design of a Squaring Architecture based on Yavadunam Sutra of Vedic Mathematics," _2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 2020, pp. 1162-1167, doi: 10.1109/ICESC48915.2020.9155768

[5]. Bindu Swetha Pasuluri, VJK Kishor Sonti, "Reducing Computational Complexity in Digital Circuit Designing using Ancient Mathematics: A Review", 2022 IEEE Delhi Section Conference (DELCON), 20 April 2022, DOI: 10.1109/DELCON54057.2022.9753237.

[6]. A. Gupta, P. Kumawat and G. Kaur, "An Efficient Approach to Implement Multiplier using Vedic Maths Squaring Technique," 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2019, pp. 90-93.

[7]. D. K. Kahar and H. Mehta, "High speed vedic multiplier used vedic mathematics," 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2017, pp. 356-359, doi: 10.1109/ICCONS.2017.8250742.

[8]. S. Akhter and S. Chaturvedi, "Modified Binary Multiplier Circuit Based on Vedic Mathematics," 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2019, pp. 234-237, doi: 10.1109/SPIN.2019.8711583

[9]. "Vedic Mathematics - Sixteen Simple mathematical formulae from the Vedas for one-line answers to all mathematical problems" By Jagadguru Shankaracharya. (Banaras Hindus University). Published online by Cambridge University Press: 22 November 2016.

[10]. Behrooz Parhami. 2010. Computer Arithmetic: Algorithms and Hardware Designs. Oxford University Press, New York, NY.

[11]. Faraji, S. Rasoul, Pierre Abillama, and Kia Bazargan. "Approximate constant-coefficient multiplication using hybrid binary-unary computing for FPGAs." ACM Transactions on Reconfigurable Technology and Systems (TRETS) 15, no. 3 (2021): 1-25.

[12]. Dani, S. G. "Myths and reality, on "Vedic mathematics"." Frontline 10, no. 21 (1993): 90-92.

 This work is licensed under a Creative Commons Attribution 4.0 International License