e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

EFFICIENT ALU DESIGN WITH PIPELINE OPTIMIZATION BASED ON VEDIC AND MODULAR MULTIPLICATION TECHNIQUES

Alpana Nekhar*1, Divyanshu Rao*2, Ravi Mohan*3

*1,2,3Shri Ram Institute Of Technology, India.

ABSTRACT

The arithmetic and Logic Unit (ALU) is the most crucial and core component of the central processing unit as well as of several embedded systems and microprocessors. ALU consists of many computational units like adders, multipliers, logical units etc. Vedic Mathematics concepts are proposed here for designing the computational units of an 8-bit ALU. Here, a high-speed 8×8bit multiplier is proposed which is based on the Vedic multiplier mechanism. A divider based on Vedic mathematics is also proposed here. The proposed Vedic mathematics-based ALU is designed using high-level hardware description language – Verilog, followed by synthesisation using the EDA tool, Xilinx ISE 14.1. Finally, the synthesized circuit has been implemented on Xilinx Spartan-6 Field Programmable Gate Array (FPGA) device.

Keywords: Integrated Software Environment, Hardware Description Language, Modular Architecture, Vedic Computing.

I. INTRODUCTION

The Central Processing Unit's main Arithmetic Logic Unit (ALU) is responsible for numerous arithmetic and logical operations. The speed of the number-crunching unit is of outrageous significance and relies enormously upon the speed of the multiplier. In this manner, the advances are continuously searching for new calculations and equipment to carry out this activity in a much- improved manner in terms of region and speed. The various subfields of mathematics, including arithmetic, algebra, geometry, and others, are the focus of Vedic Mathematics. The utilization of Vedic Arithmetic ALUs in the calculation of a processor will diminish the intricacy of execution time, region and power utilization and so forth.

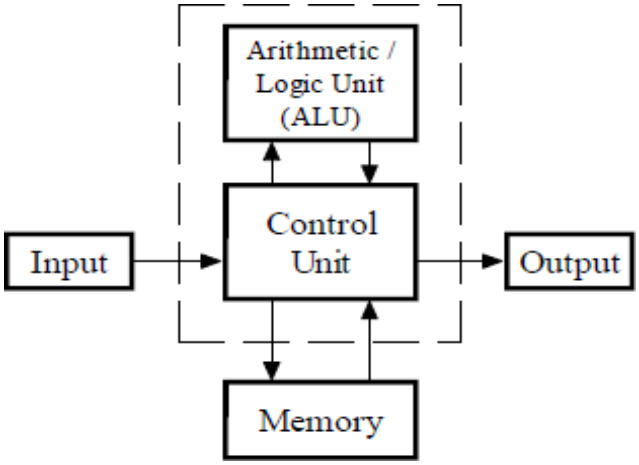

Fig 1: Basic block diagram of a computer system

II. LITERATURE REVIEW

Milam Gadda et al [2] The paper gives the subtleties of a 64-cycle ALU configuration given Vedic Sutras like Urdhva Tiryakbhyam and Nikhilam and execution results on FPGA. The planned Vedic multiplier gets results after estimation in 4.014ns time. Xilinx's Simple FPGA pack is used for understanding the total ALU module which is coded utilizing Verilog HDL. Literature Review In this paper, Chiranjeevi G.N. et al. [1] discuss a design architecture that uses the Vedic sutra to perform mathematical operations for upward compatibility in a pipeline manner. Notwithstanding expanding the region, execution and diminishing power, Vedic engineering has been seen to be intrinsically viable with higher productivity for pipeline design. However, Vedic architecture enables pipelines to be compactable to any base modules for any length, starting with 2-bit base modules and progressing to 8-bit modules (L=1,2,3)

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

Table 1: Literature review

Title	author	Outcomes
Pipeline Architecture for N=K*2L Bit Modular ALU: Case Study between Current Generation Computing and Vedic Computing	chiranjeevi G.N et al [1], IEEE-2021	authors have extensively verified modular architecture for 4-Bit modules for 16-bit and 32 bit pipelined operations. Individually multiplication using Urdhva Tiryakbhyam, division using Dhwajanka sutra, modulo using Dwandwayoga sutra. 72 LUT and 15.35 ns time delay for 16 bits for modulo.
64-bit ALU Design using Vedic Mathematics	nilam Gadda et al [2], IEEE-2020	Paper provides the details of a 64-bit ALU design based on Vedic Sutras like Urdhva Tiryakbhyam and Nikhilam and implementation results on FPGA. The designed Vedic multiplier obtains results after calculation in 4.014ns time, for 8x8 multiplication.
Improved Modulo (2^n + 1) Multiplier	Kalyani Palutla et al [3], springer-2020	a new modulo 2^n+1 multiplier is proposed for ALU block cipher complemented carry followed by modulo adder. 44.46 ns time delay for 2^{16}+1 modulo operation and

III. DESIGN METHODOLOGY

ALU is the key element of Microprocessors, Microcontrollers, and embedded systems. Figure below explains the component of ALU. Modulo is one of the prime requirements of any ALU designing, as we know ALU can perform addition, subtraction, modulo and all logical operations, a combination of addition, subtraction & modulo is known as MAC (Multiply and Accumulate) in modern Microprocessors & Microcontroller, the method which we choose for designing MAC will affect Microprocessor or Microcontroller performance.

Modulo Techniques: The numbers 1,4,9, 16 are called Modulo Numbers because you can arrange the number of counters to form

a Modulo. The 4 Counters are in 2 rows of 2. The 9 counters are in 3 rows and 3 columns. 1 x 1 = 1, 2 x 2 = 4, 3 x 3 =9 , 4x4 = 16, So if we modulo a number we multiply it by itself. 3 Modulod is (32) =9 ; 4 Modulo is (42) =16; Modulo numbers always have an odd number of factors. All other numbers have an even number of factors.

Many researchers proposed arithmetic algorithms at simulation level using vedic sutra. These algorithms have been evaluated with better performance, area and speed. The literature has been widely found to be towards individual arithmetical operators like multiplier, modulo and cube and so on. A consolidated computing architecture, especially Nbit ALU is yet be realized Generalized N-bit ALU, can always be realized using pipeline modular architecture. The proposition is on realizing N-bit using '2L ' as base modules using 'K' modules in pipelining. The authors have extensively verified modular architecture for 4-Bit modules for 16 bit and 32 bit pipelined operations. Individually multiplication using Urdhva Tiryakbhyam, division using Dhwajanka sutra, modulo using Dwandwayoga sutra. MAC unit which involves multiplication algorithms used in FFT and IFFT using sutras of Vedic mathematics and it is possible to achieve reduce version interms of speed and delay, compared to different generations of ALU The authors are now exploring N-bit ALU architecture FPGA implementation using Vedic sutras with flexible modular pipeline architecture and mainly targeted for Digital Signal processing applications

Diminished-1 Method: In the Diminished-1 number system, the number A is represented by A'=A-1 and the value zero is treated separately, i.e., it requires an additional zero indication bit.

S'= (S-1) = (X+Y-1) mod (2n)

= [(X'+1) + (Y'+1) - 1)] mod (2n)

= (X' + Y'+ 1) mod (2n)

$$(X'+Y'+1)\bmod(2^n+1)=\begin{cases}X'+Y'+1-(2^n+1)=(X'+Y')\bmod 2^n & \text{if } X'+Y'+1\geq 2^n\\ X'+Y'+1 & \text{otherwise}\end{cases}$$

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

The above equation can be depicted by,

Algorithm 1 (Modulo 2n +1 addition in diminished-1 number system): A number in Diminished-1 is represented by n bits. The nth bit is used to indicate '0'. The modulo 2n addition algorithm has been presented for zero and non zero operands:

If the most significant bit of one addend is '1', inhibit the addition and the other addend is the output.

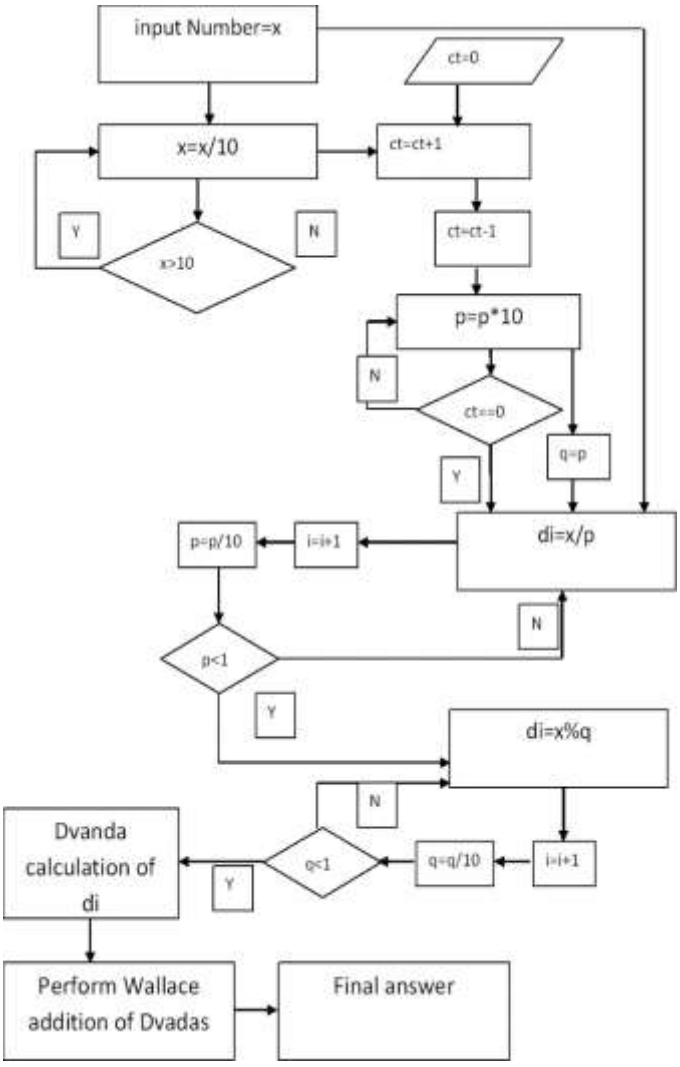

Fig 2: Proposed Modulo Method work flow

If the MSB of both addends are '0', ignore the MSB, add the n lsb's, complement the carry and add it to the n lsb's of the sum.

One structure of diminished-1 addition algorithm is depicted in Figure. The proposed Arithmetic Module has first been split into three smaller modules (shown in fig 1), that is

1. Multiplier
2. MAC unit
3. Arithmetic module,

As a whole. These modules have been made using Verilog HDL and simulated for different case study and synthesized in Xilinx Vivado 2016.1 with Spartan 3 family and XC3S400 device and Zynq700 Zedboard as target. Arithmetic block is considered as unique and important functional block. It handles all the Arithmetic and logic operations that are required for user constraints.in this proposed method arithmetic block is implemented using Vedic algorithms and priority is given for multiplication. To implement addition and subtraction conventional method is used Design starts with the implementation of multiplier design of size 2x2(21*1), where it is equivalent to 2L* K bit multiplier

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

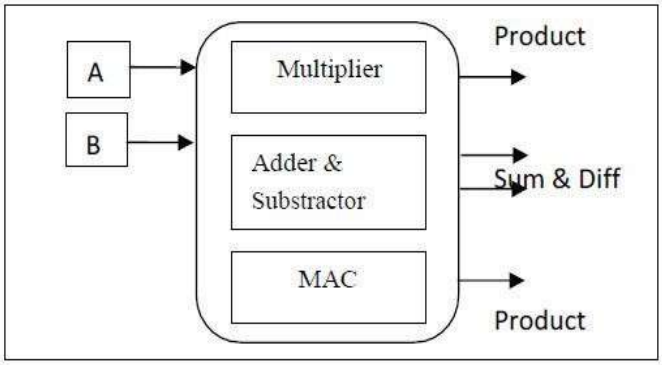

Fig 3: Basic block diagram of Arithmetic unit

IV. RESULT AND COMPRESSION

Synthesis Results: Figure 4(a) shows the synthesis result of modulo multiplier in which total 47 slices are used and total 4656 slices are available. In this design 23 bonded IOBs are used and 158 IOBs are available.

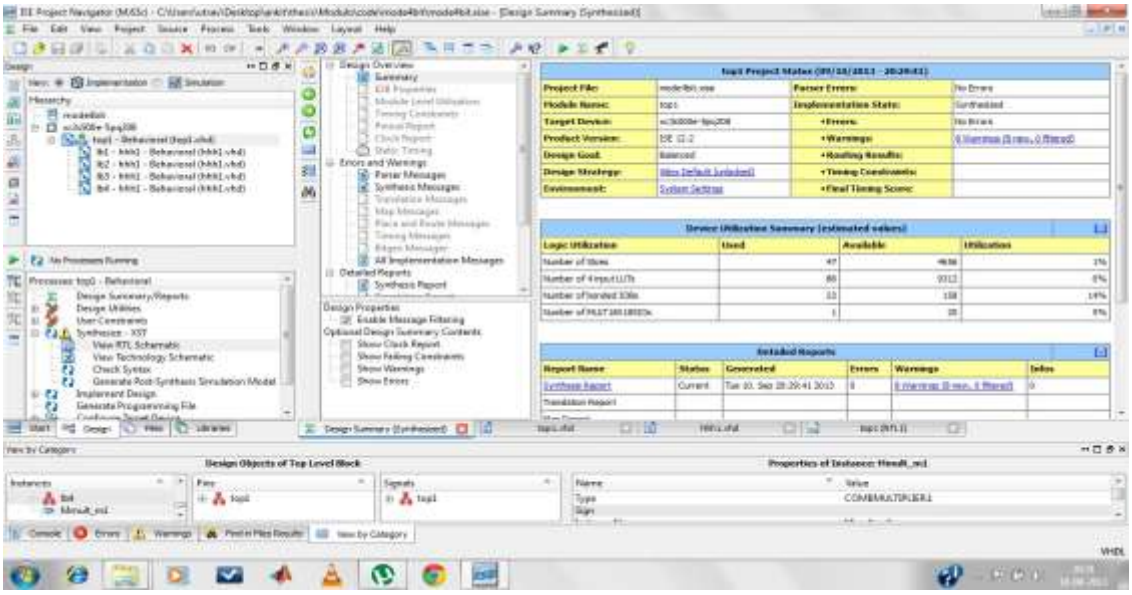

Figure 4(a): Synthesis Result of Modulo 2^4+1 Multiplier

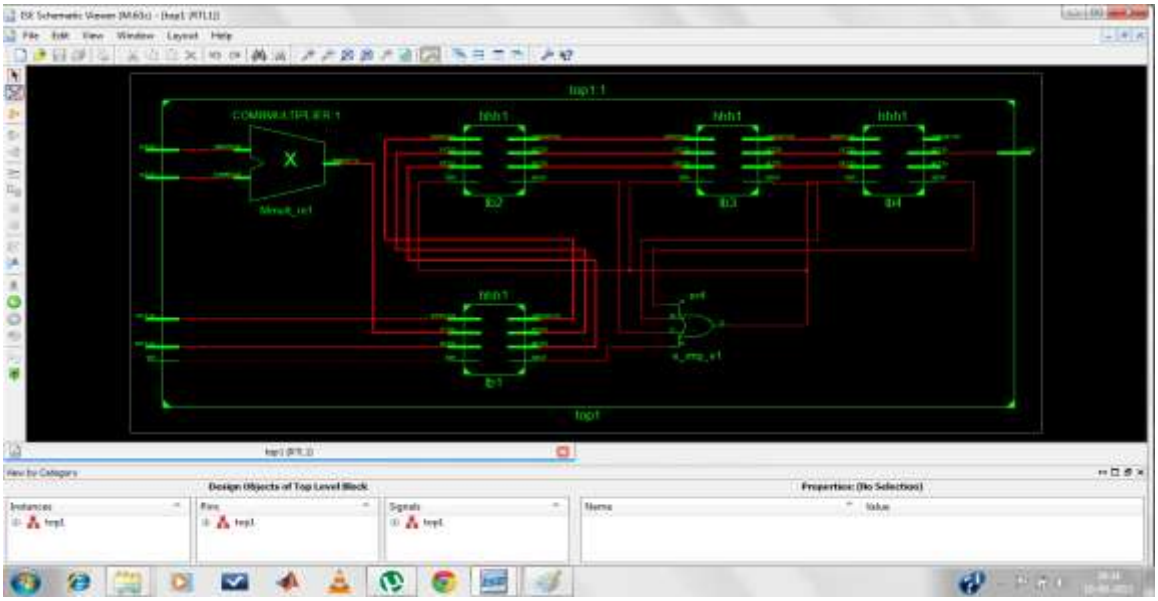

Figure 4(b): Schematic Diagram of modulo 2^4+1 multiplier

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

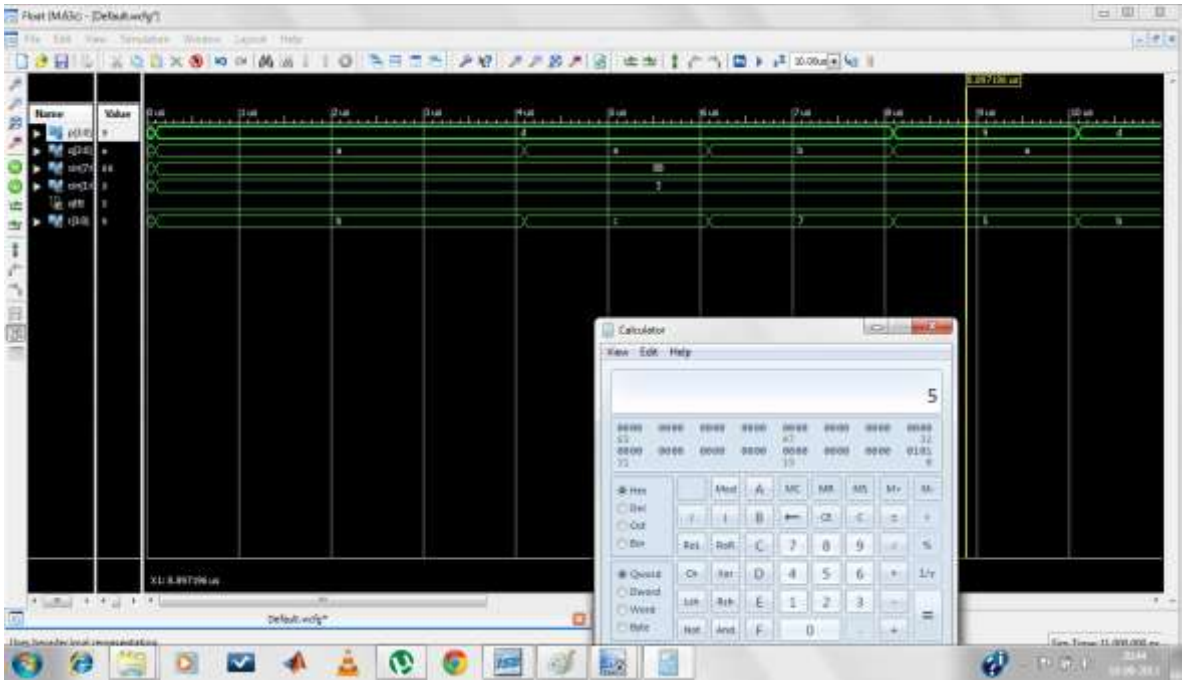

Figure 4(c) Simulation of modulo 2^4+1 multiplier

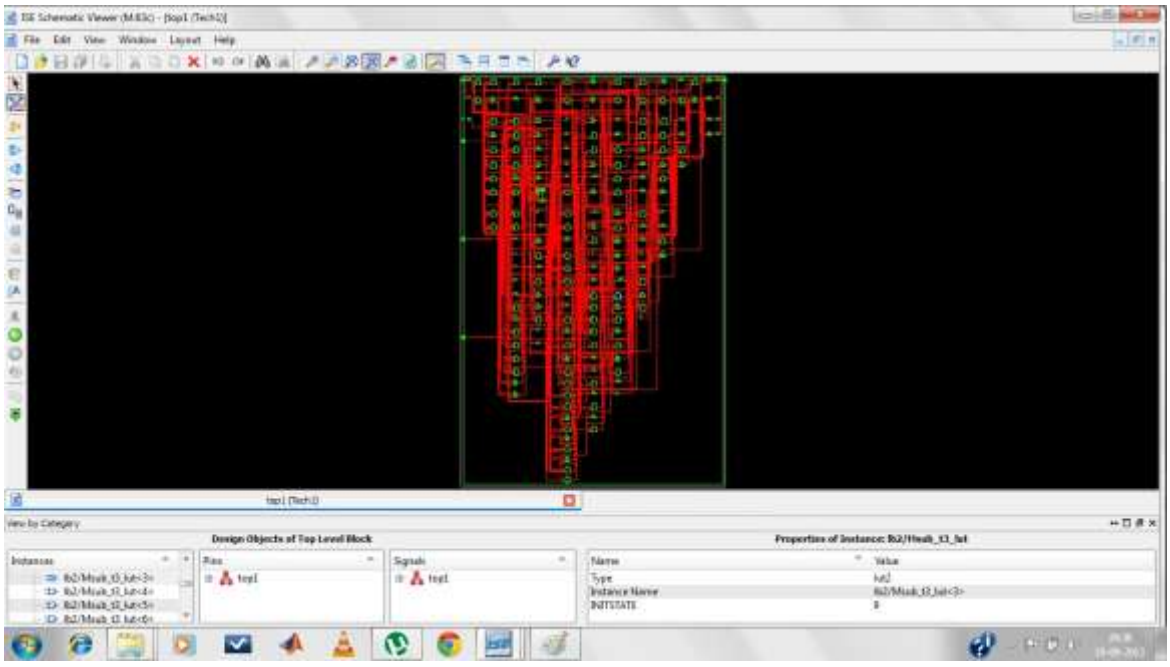

Figure 4(d): Schematic Viewer of modulo 2^4+1 multiplier

Table 2: Comparative results

author	Work	Proposed Work Modification
chiranjeevi G.N et al [1]	modular architecture for 4-Bit modules for 16 bit and 32-bit pipelined operations. Individually multiplication using Urdhva Tiryakbhyam, division using Dhwajanka sutra, modulo using Dwandwayoga sutra. 72 FPGA slice used and 15.35 ns time delay for 2^4+1 modulo multiplication.	This work has design ALU with the help of modulo operation, where multiplication part is done using Urdhva Tiryakbhyam sutra. 47 slices are used for 2^n+1 multiplication. Time delay for 24+1 modulo multiplication is 16.340 ns.
nilam Gadda et	Paper provides the details of a 64-bit ALU design based on Vedic Sutras like Urdhva Tiryakbhyam.	This work has design of 64-bit ALU based on Vedic Sutras like Urdhva Tiryakbhyam.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

al [2]	designed Vedic multiplier obtains results after calculation in 4.014ns time, for 4x4 multiplication.	And the time delay for the 4x4 multiplication found is 3.872 ns.
Kalyani Palutlaet al [3]	This work is deigned of modulo multiplication for ALU encryption, 44.46 ns time delay for $2^{16}+1$ modulo operation.	Proposed work $2^{16}+1$ modulo operation uses 1217 vertex-5 FPGA slice and time delay of 43.19 ns.

V. CONCLUSION

Vedic Math ALUs for different numerical calculations are ended up being effective as contrasted with customary strategies. Numerous scientists have proposed computational units in view of Vedic Math for different sign handling applications and these plans are ended up being proficient ones. The Urdhva-Tiryakbyham Sutra and the Nikhilam Sutra are utilized in the proposed Vedic Mathematics-based ALU to perform division and multiplication, respectively. Several areas of signal processing would benefit from the proposed ALU architecture. FPGA is profitable as far as cost, advancement time, cost, viability and practicability. All FPGA designs can achieve stability and reduce the likelihood of design errors. FPGA is used to redesign out of date coordinated circuits decreases equipment circuit board changes, increments efficiency, and guarantees that the functional requirements are met.

VI. REFERENCES

[1] G. N. Chiranjeevi and S. Kulkarni, "Pipeline Architecture for N=K*2L Bit Modular ALU: Case Study between Current Generation Computing and Vedic Computing," 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2021, pp. 1-4, doi: 10.1109/I2CT51068.2021.9417917.

[2] N. Gadda and U. Eranna, "64-bit ALU Design using Vedic Mathematics," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-4, doi: 10.1109/ic-ETITE47903.2020.122.

[3] panelKalyani Palutla, Prakash Gundabathina, Implementation of High Speed Modulo (2^n+1) Multiplier for ALU Cipher, Procedia Computer Science, Volume 171, 2020, Pages 2016-2022

[4] G. N. Chiranjeevi and S. Kulkarni, "Pipeline Architecture for N=K*2L Bit Modular ALU: Case Study between Current Generation Computing and Vedic Computing," 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2021, pp. 1-4, doi: 10.1109/I2CT51068.2021.9417917.

[5] S. Lad and V. S. Bendre, "Design and Comparison of Multiplier using Vedic Sutras," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, pp. 1-5, doi: 10.1109/ICCUBEA47591.2019.9128517

[6] A. Yadav and V. Bendre, "Design and Verification of 16 bit RISC Processor Using Vedic Mathematics," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2021, pp. 759-764, doi: 10.1109/ESCI50559.2021.9396965.

[7] Ande, L. D. Kalidindi, P. K. Mallula, P. V. Dantuluri and N. Vegesna, "High-Speed Vedic Multiplier Implementation Using Memristive and Speculative Adders," 2022 International Conference on Computing, Communication and Power Technology (IC3P), Visakhapatnam, India, 2022, pp. 186-189, doi: 10.1109/IC3P52835.2022.00046.

[8] H. Chugh and S. Singh, "Design and Implementation of a High-Performance 4-bit Vedic Multiplier Using a Novel 5-bit Adder in 90nm Technology," 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2022, pp. 1-6, doi: 10.1109/ICRITO56286.2022.9964936.

[9] E. Sekar, S. Palaniswami, S. P. Philip, R. S. S, G. K and D. A, "Design of Reconfigurable Signed Dual Modulo Multiplier Function (DMMF) for RNS," 2021 Smart Technologies, Communication and Robotics (STCR), Sathyamangalam, India, 2021, pp. 1-5, doi: 10.1109/STCR51658.2021.9588822.

[10] K. Patel and J. Kanungo, "Efficient Tree Multiplier Design by using Modulo 2n + 1 Adder," 2021 Emerging Trends in Industry 4.0 (ETI 4.0), Raigarh, India, 2021, pp. 1-6,

doi: 10.1109/ETI4.051663.2021.9619220.

[11] Kouretas and V. Paliouras, "Radix-3 low-complexity modulo-M multipliers," 2019 29th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS), Rhodes, Greece, 2019, pp. 107-112, doi: 10.1109/PATMOS.2019.8862036.

[12] Paul, A. Nath, S. Krishnaswamy, J. Pidanic, Z. Nemec and G. Trivedi, "Tensor Based Multivariate Polynomial Modulo Multiplier for Cryptographic Applications," in IEEE Transactions on Computers, vol. 72, no. 6, pp. 1581-1594, 1 June 2023, doi: 10.1109/TC.2022.3215638.

[13] P. S. Phalguna, D. V. Kamath and P. V. A. Mohan, "Design of multipliers for moduli$({2^{4k}} + {2^{2k + 1}} - {2^{3k + 1}} - {2^k} + 1)$ and $({2^{4k}} - {2^{3k + 1}} + {2^k} - 1)$," 2021 International Conference on Circuits, Controls and Communications (CCUBE), Bangalore, India, 2021, pp. 1-6, doi: 10.1109/CCUBE53681.2021.9702727.

[14] A. T, S. S, R. A and S. K M, "FPGA –based Optimized Design of Montgomery Modular Multiplier using Karatsuba Algorithm," 2023 Second International Conference on Electronics and Renewable Systems (ICEARS), Tuticorin, India, 2023, pp. 131-135, doi: 10.1109/ICEARS56392.2023.10085256.