International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 3 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

Posit-Based Arithmetic Unit Design Using HUB Number Representation

[1]A. Ganga Sagar,[2]M. Bharsharani Dora,[3]M. Bhanu, G Deepthi[*]

[1,2,3]B. Tech Students, Department of Electronics and Communication Engineering, Samskruti College Of Engineering & Technology

[*] Department of Electronics and Communication Engineering, Samskruti College Of Engineering & Technology

[*]Email:deepthigurijala23@gmail.com

ABSTRACT

The rapid advancement of digital arithmetic architectures has increased the demand for high-speed, low-power, and accuracy-efficient numerical computing systems. Traditional binary arithmetic units often suffer from limitations such as rounding errors, overflow, underflow, and high hardware complexity when used in scientific computing, signal processing, embedded systems, and artificial intelligence applications. To overcome these limitations, the proposed HUB Meets Posit: Arithmetic Units Implementation introduces an efficient arithmetic architecture based on the Posit number system. The Posit format provides better dynamic range, improved numerical accuracy, and reduced hardware overhead compared to conventional floating-point representations. The proposed system implements arithmetic units such as addition, subtraction, multiplication, and division using posit-based computation techniques. It integrates decoding, regime extraction, exponent processing, fraction operation, normalization, and encoding modules to perform accurate arithmetic operations. The implementation improves computational precision, reduces memory usage, and enhances performance for hardware-based numerical applications. By combining posit arithmetic with optimized hardware design methodologies, the proposed architecture provides an efficient alternative for modern digital processing systems. The system is highly suitable for machine learning accelerators, digital signal processing, embedded processors, scientific computing, and low-power hardware implementations.

Keywords: Posit Arithmetic, HUB, Arithmetic Units, Posit Number System, Digital Arithmetic, Floating Point Alternative, VLSI Design, Hardware Implementation, Low Power Computing, Numerical Accuracy, Arithmetic Logic Unit, FPGA, Digital Signal Processing, Embedded Systems, Computer Architecture.

I. INTRODUCTION

The continuous growth of data-intensive computing has increased the demand for arithmetic systems that can provide high numerical accuracy without excessive hardware cost. Conventional floating-point formats are widely used in digital processors, but their fixed allocation of exponent and fraction bits may not always provide the best balance between dynamic range, precision, power consumption, and circuit complexity. Posit arithmetic has therefore gained considerable attention as an alternative numerical representation for modern computing platforms. Studies have shown that posit numbers can offer competitive numerical accuracy and computational efficiency, particularly when suitable word sizes and exponent configurations are selected [1].

The hardware realization of posit arithmetic is an important area of research because the benefits of a number representation depend heavily on the efficiency of its underlying arithmetic units.

Multiply-accumulate operations, for example, are fundamental to signal processing, machine learning, and scientific workloads. Energy-efficient fused posit MAC architectures have demonstrated that intermediate rounding can be reduced while maintaining useful numerical accuracy [2]. Similarly, configurable hardware generators have been introduced to simplify the development and FPGA prototyping of posit arithmetic circuits with different design parameters [3].

The use of posit arithmetic has also expanded into processor-level architectures. Lightweight posit processing units integrated with RISC-V systems have demonstrated the potential of this number format for deep neural network workloads [4]. Other studies have investigated posit arithmetic for the training and deployment of generative adversarial networks, indicating that reduced-precision numerical formats can support complex artificial intelligence computations while lowering hardware requirements [5]. Posit-based computation has additionally been evaluated in optical-flow processing, where the relationship between numerical quality and implementation efficiency is particularly important [6].

Processor customization has further strengthened the practical relevance of posit arithmetic. The integration of posit and quire operations into the CVA6 RISC-V core has shown that alternative arithmetic formats can be incorporated into general-purpose processor architectures [7]. At the arithmetic-unit level, dedicated posit adders have been developed to address regime processing, operand alignment, normalization, and rounding operations [8]. Verilog-based implementations of posit adders and multipliers have also provided a foundation for evaluating such units on digital hardware platforms [9].

Despite these developments, posit arithmetic still involves additional encoding and decoding operations caused by its variable-length regime field. This can increase the complexity of arithmetic circuits. A promising approach to reducing this overhead is the combination of posit arithmetic with the Half-Unit-Biased (HUB) representation. HUB-based processing can simplify certain rounding and arithmetic operations by using an implicit least significant bit. The combination of HUB and posit concepts has demonstrated opportunities for developing arithmetic units with reduced circuit complexity and improved implementation efficiency [10].

Recent research has continued to improve the hardware characteristics of posit computation. Energy-efficient multiplier architectures have been proposed to reduce the power and area requirements of multiplication-intensive systems [11]. At the other end of the precision range, 64-bit posit processing has been investigated for scientific computing, demonstrating the scalability of posit arithmetic beyond low-precision embedded applications [12]. Accelerator-based evaluations have further examined the practical behaviour of posit operations in high-performance computing environments [13]. Specialized arithmetic functions, including square-root units with reduced iteration requirements, also show that posit hardware research is progressing beyond basic addition and multiplication [14].

The growing body of work on posit principles, hardware architectures, and application domains confirms the increasing importance of alternative numerical systems in energy-aware computing [15]. However, there remains a need for arithmetic architectures that retain the numerical advantages of posit representation while simplifying hardware operations. Therefore, this work focuses on the design of posit-based arithmetic units using HUB number representation. The proposed approach aims to explore efficient addition and multiplication architectures with emphasis on hardware complexity, computational performance, resource utilization, and suitability for FPGA or VLSI implementation.

II. LITERATURE SURVEY

S. D. Ciocirlan et al. (2021) Ciocirlan and co-authors studied the accuracy and computational efficiency of posit arithmetic as an alternative to conventional IEEE 754 floating-point representation. The authors developed a flexible posit arithmetic unit and evaluated different posit configurations using mathematical, machine learning, and benchmark applications. Their results indicated that properly selected posit formats can provide competitive accuracy

while reducing numerical storage requirements. The study also highlighted that hardware cost varies with posit size and configuration, making efficient arithmetic-unit design an important research requirement.

R. Murillo et al. (2021) Murillo and co-authors proposed energy-efficient multiply-accumulate units based on fused posit arithmetic. The architecture used quire-supported accumulation to reduce repeated rounding during successive multiplication and addition operations. The work demonstrated that fused processing can improve numerical behaviour while providing design alternatives for systems requiring either reduced hardware area or higher computational throughput. This research is particularly relevant to posit-based arithmetic units used in signal processing and machine learning applications.

D. Shekhawat et al. (2021) Shekhawat, Upadhyay, and Pandey presented a configurable hardware generator for posit arithmetic along with FPGA prototyping. Their approach focused on simplifying the development of posit-based arithmetic circuits with different word lengths and exponent configurations. The generated hardware enabled designers to examine implementation characteristics such as resource utilization and processing performance. This work provided an important foundation for developing customizable posit arithmetic units suitable for FPGA-based systems.

M. Cococcioni et al. (2021) Cococcioni and co-authors developed a lightweight posit processing unit intended for integration with RISC-V processors and deep neural network applications. The study explored how posit arithmetic could support computationally intensive AI workloads while maintaining a compact hardware architecture. Their findings demonstrated the practical possibility of incorporating alternative numerical formats into embedded processor platforms, supporting further research into efficient posit arithmetic accelerators.

D. Mallasén, A. A. Del Barrio, and M. Prieto-Matias (2022) Mallasén and co-authors customized the CVA6 RISC-V processor core by integrating posit and quire arithmetic support. Their work showed that posit operations could be introduced into a general-purpose processor architecture rather than being restricted to isolated arithmetic modules. The inclusion of quire arithmetic supported accurate accumulation operations and strengthened the suitability of posit representation for scientific and data-intensive computations.

R. Murillo et al. (2024) Murillo and co-authors introduced an arithmetic architecture that combines the Half-Unit-Biased (HUB) representation with the posit number system. The proposed posit-HUB format was applied to adder and multiplier implementations to simplify specific arithmetic and rounding operations. The study demonstrated that the HUB concept can reduce hardware complexity while preserving the main numerical characteristics of posit arithmetic. This work provides the strongest foundation for developing posit-based arithmetic units using HUB number representation.

III. EXISTING SYSTEM

In the existing digital computing systems, arithmetic operations are commonly performed using binary integer and IEEE floating-point formats. Floating-point arithmetic is widely used in processors and hardware accelerators because it supports real-number computation. However, floating-point units require complex hardware structures for exponent alignment, mantissa operation, rounding, normalization, and exception handling.

The existing floating-point arithmetic systems suffer from several limitations such as high power consumption, large area requirement, rounding errors, overflow, underflow, and reduced accuracy in certain numerical ranges. These limitations become more significant in applications such as machine learning, scientific computing, signal processing, and embedded systems where large numbers of arithmetic operations are performed continuously.

Traditional arithmetic units also distribute precision inefficiently because the exponent and mantissa fields are fixed. This results in unnecessary loss of precision for small and medium-range values. In addition, floating-point hardware requires additional logic for special cases such as NaN, infinity, zero, and denormal numbers, increasing circuit complexity.

Due to these drawbacks, there is a need for an improved arithmetic architecture that provides better numerical accuracy, reduced hardware complexity, and efficient implementation for modern digital systems.

IV. PROPOSED SYSTEM

The proposed system implements arithmetic units using the Posit number system to improve computational accuracy and hardware efficiency. The architecture supports basic arithmetic operations such as addition, subtraction, multiplication, and division using posit-based data representation. The system accepts posit input operands, decodes them into internal fields, performs arithmetic processing, normalizes the output, and encodes the final result back into posit format.

The proposed architecture consists of a posit decoder, sign extraction unit, regime detector, exponent extraction unit, fraction processing unit, arithmetic operation unit, normalization unit, rounding unit, and posit encoder. The posit decoder separates the input posit number into sign, regime, exponent, and fraction components. These components are then processed according to the selected arithmetic operation.

For addition and subtraction operations, the system performs operand comparison, scale alignment, fraction addition or subtraction, normalization, and result encoding. For multiplication and division operations, the system performs sign calculation, regime and exponent computation, fraction multiplication or division, normalization, and final posit generation.

The proposed posit arithmetic units provide better precision and dynamic range compared to conventional floating-point arithmetic. The architecture reduces hardware overhead by eliminating several complex floating-point exception-handling mechanisms. It also improves energy efficiency and performance, making it suitable for FPGA and ASIC implementation.

The proposed system is highly suitable for machine learning accelerators, embedded processors, digital signal processing systems, scientific computing modules, and low-power arithmetic hardware applications.

V. BLOCK DIAGRAM

The system architecture of the proposed **HUB Meets Posit Arithmetic Units Implementation** consists of multiple functional modules such as input posit operands, posit decoder, arithmetic control unit, operation selection unit, arithmetic processing blocks, normalization unit, rounding unit, posit encoder, and output result register.

The input module receives posit operands from memory or processor registers. These operands are passed to the posit decoder, which extracts the sign bit, regime value, exponent field, and fraction field. The decoded values are converted into an internal representation suitable for arithmetic computation.

The arithmetic control unit selects the required operation such as addition, subtraction, multiplication, or division based on the control signal. In the arithmetic processing stage, separate hardware blocks perform posit addition, posit subtraction, posit multiplication, and posit division. These units process the decoded operands and generate intermediate results.

The normalization unit adjusts the result to match the posit format and ensures correct scale representation. The rounding unit improves numerical accuracy by rounding the computed value according to the posit encoding rules. Finally, the posit encoder combines the sign, regime, exponent, and fraction fields to generate the final posit output.

The result register stores the final arithmetic output and sends it to the processor, memory, or connected computing module. The complete architecture provides high-speed, accurate, and hardware-efficient arithmetic computation suitable for modern digital systems.

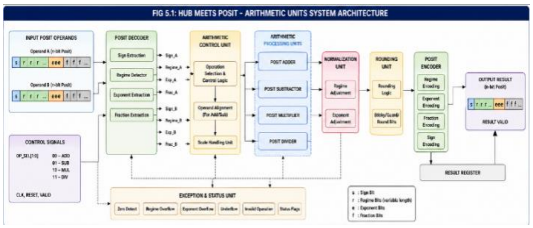


Fig 5.1: Posit Arithmetic Units System Architecture

VI. IMPLEMENTATION

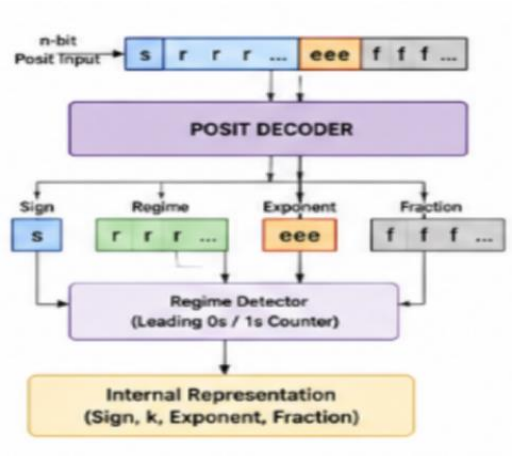


Fig 6.1: Posit Decoder and Field Extraction Unit

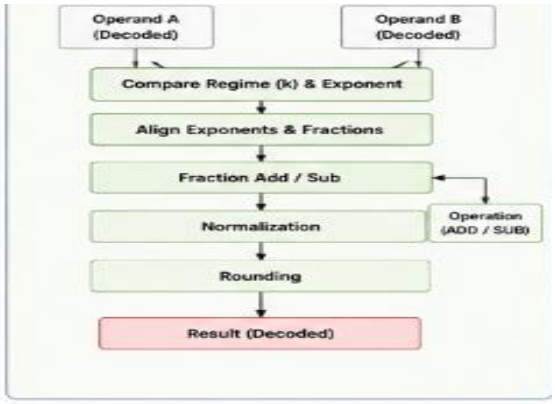


Fig 6.2: Posit Addition and Subtraction Unit

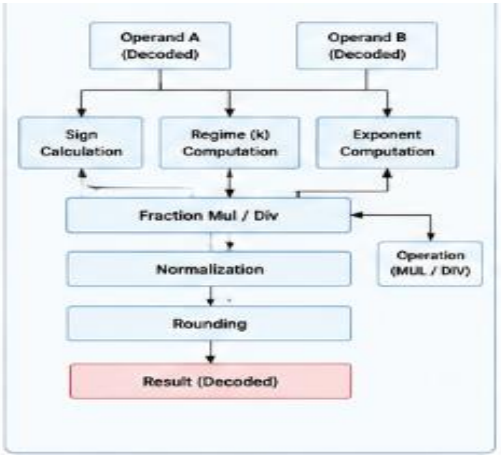


Fig 6.3: Posit Multiplication and Division Unit

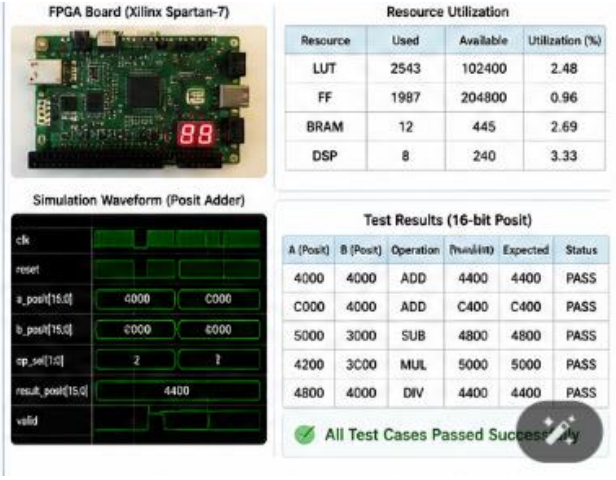


Fig 6.4: FPGA-Based Posit Arithmetic Hardware Testing

VII. CONCLUSION

The proposed **HUB Meets Posit: Arithmetic Units Implementation** provides an efficient arithmetic architecture for modern digital computing systems. By using the Posit number system, the proposed design improves numerical accuracy, dynamic range, and computational efficiency compared to traditional floating-point arithmetic units. The architecture supports addition, subtraction, multiplication, and division operations using optimized posit decoding, processing, normalization, and encoding techniques.

The system reduces hardware complexity, improves precision, and provides better energy efficiency for FPGA and ASIC-based implementations. The proposed posit arithmetic units are suitable for machine learning accelerators, embedded systems, digital signal processing, scientific computing, and low-power numerical hardware applications. Thus, the proposed system serves as a reliable and efficient arithmetic solution for next-generation computing architectures.

VIII. FUTURE SCOPE

The proposed posit arithmetic architecture has significant scope for future enhancement in advanced computing systems. In the future, the design can be extended to support configurable posit formats such as 8-bit, 16-bit, 32-bit, and 64-bit posit arithmetic for different application

requirements.

Advanced optimization techniques can be applied to reduce area, delay, and power consumption further. The architecture can also be integrated with machine learning accelerators, neural network processors, and digital signal processing chips to improve computational accuracy and energy efficiency.

Future work may include the development of a complete posit-based Arithmetic Logic Unit (ALU), posit-based processor core, and hybrid floating-point-posit computing system. The design can also be implemented using ASIC technology for high-speed and low-power applications.

Additionally, software-hardware co-design methodologies can be used to improve compatibility with modern programming environments. Overall, posit arithmetic provides a promising direction for future high-performance, accurate, and energy-efficient computing systems.

IX. REFERENCES

[1] S. D. Ciocirlan, D. Loghin, L. Ramapantulu, N. Ţăpuş, and Y. M. Teo, "The Accuracy and Efficiency of Posit Arithmetic," in 2021 IEEE 39th International Conference on Computer Design (ICCD), 2021, pp. 83–87.
DOI: 10.1109/ICCD53106.2021.00024.

[2] R. Murillo, A. A. Del Barrio, and G. Botella, "Energy-Efficient MAC Units for Fused Posit Arithmetic," in 2021 IEEE 39th International Conference on Computer Design (ICCD), 2021.
DOI: 10.1109/ICCD53106.2021.00032.

[3] D. Shekhawat, H. Upadhyay, and J. G. Pandey, "A Hardware Generator for Posit Arithmetic and its FPGA Prototyping," in 2021 25th International Symposium on VLSI Design and Test (VDAT), 2021.
DOI: 10.1109/VDAT53777.2021.9601025.

[4] M. Cococcioni, F. Rossi, E. Ruffaldi, and S. Saponara, "A Lightweight Posit Processing Unit for RISC-V Processors in Deep Neural Network Applications," IEEE Transactions on Emerging Topics in Computing, 2021.
DOI: 10.1109/TETC.2021.3120538.

[5] A. A. Del Barrio, R. Murillo, and G. Botella, "Posit Arithmetic for the Training and Deployment of Generative Adversarial Networks," in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2021.
DOI: 10.23919/DATE51398.2021.9473933.

[6] V. Saxena, S. P. Singh, and S. K. Nandy, "Brightening the Optical Flow through Posit Arithmetic," in 2021 22nd International Symposium on Quality Electronic Design (ISQED), 2021.
DOI: 10.1109/ISQED51717.2021.9424360.

[7] D. Mallasén, A. A. Del Barrio, and M. Prieto-Matias, "Customizing the CVA6 RISC-V Core to Integrate Posit and Quire Arithmetic," in 2022 XXXVII Conference on Design of Circuits and Integrated Systems (DCIS), 2022.
DOI: 10.1109/DCIS55711.2022.9970026.

[8] S. N. S. and A. S. B. P., "Implementation of Regime-5 Posit Adder," in 2022 Third International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), 2022, pp. 1040–1043.
DOI: 10.1109/ICICICT54557.2022.9917949.

[9] R. Sanivarapu and Y. Mallikarjuna Rao, "Design and Implementation of POSIT Based Adder and Multiplier in Verilog HDL," E3S Web of Conferences, vol. 391, Art. no. 01184, 2023.
DOI: 10.1051/e3sconf/202339101184.

[10] R. Murillo, A. A. Del Barrio, and G. Botella, "HUB Meets Posit: Arithmetic Units Implementation," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 71, no. 1, pp. 440–444, 2024.
DOI: 10.1109/TCSII.2023.3307488.

[11] A. A. Jonnalagadda et al., "Design of Energy Efficient Posit Multiplier," in Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI), 2023.
DOI: 10.1145/3583781.3590227.

[12] D. Mallasén, A. A. Del Barrio, and M. Prieto-Matias, "Big-PERCIVAL: Exploring the Native Use of 64-Bit Posit Arithmetic in

Scientific Computing," IEEE Transactions on Computers, 2024.
DOI: 10.1109/TC.2024.3377890.

[13] T. Nakata et al., "Evaluation of POSIT Arithmetic with Accelerators," in Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region Workshops, 2024.
DOI: 10.1145/3635035.3635046.

[14] R. Murillo, A. A. Del Barrio, and G. Botella, "Square Root Unit with Minimum Iterations for Posit Arithmetic," in 2024 IEEE 31st Symposium on Computer Arithmetic (ARITH), 2024.
DOI: 10.1109/ARITH61463.2024.00030.

[15] D. Mallasén et al., "Navigating Posit Arithmetic: A Comprehensive Survey of Principles, Hardware, and Applications," ACM Computing Surveys, 2025.
DOI: 10.1145/3772284