ICSPIS

11th International Conference on Signal Processing & Intelligent Systems
Mazandaran University of Science and Technology, December 23-24, 2025

Mazandaran University of Science and Technology

# Low-Precision POSIT Arithmetic for Spiking Neural Networks with Kahan Summation

Shayan Naghizadeh
*Computer Engineering Department*
*Amirkabir Uiniversity of Technology*
*(Tehran Polytechnic)*
Tehran, Iran
shayan.naghizadeh@aut.ac.ir

Reza Adinepour *
*Computer Engineering Department*
*Amirkabir Uiniversity of Technology*
*(Tehran Polytechnic)*
Tehran, Iran
adinepour@aut.ac.ir

Morteza Saheb Zamani
*Computer Engineering Department*
*Amirkabir Uiniversity of Technology*
*(Tehran Polytechnic)*
Tehran, Iran
szamani@aut.ac.ir

**Abstract—Spiking Neural Networks (SNNs) are highly suitable for energy-efficient neuromorphic computing due to their event-driven nature. Unlike Artificial Neural Networks (ANNs), where POSIT arithmetic has been used for storage saving and require decoding and encoding during each operation, SNNs process information only when events occur, minimizing the overhead. This makes POSIT arithmetic particularly advantageous for SNNs, as it avoids unnecessary operations. However, at low bit-widths such as 6 bits, accumulation errors can arise, affecting accuracy. To address this problem, we propose integrating Kahan's compensated summation algorithm into the synaptic accumulation process of neurons in SNNs, which restores loss precision and enhances the stability of low-precision SNNs. Experimental results demonstrate that the proposed 6-bit POSIT format achieves 93.2% accuracy—close to 8-bit POSIT (95.1%) and 32-bit floating point (97.6%)—while reducing LUT usage by about 27% and FF usage by about 55% compared to the 32-bit floating-point implementation, highlighting its efficiency for low-power, resource-constrained neuromorphic hardware and demonstrating its potential for next-generation neuromorphic accelerators.**

***Keywords— Spiking neural network, POSIT arithmetic, low-precision computing, Kahan, neuromorphic hardware.***

## I. INTRODUCTION

Spiking neural networks have attracted increasing attention as a biologically inspired and energy-efficient alternative to conventional artificial neural networks. Unlike dense and clock-driven architectures, neurons in an SNN communicate through discrete spike events, allowing computation and memory access to occur only when necessary. This event-driven paradigm enables significant reduction in power consumption and makes SNNs highly suitable for low-power neuromorphic processors and edge intelligence applications [2], [4].

Despite these advantages, the deployment of SNNs in hardware has been limited by the high computational cost of synaptic accumulation and membrane potential updates. These operations dominate energy consumption and are sensitive to numerical precision. To address this issue, recent studies have explored low-precision arithmetic formats such as fixed-point and block floating-point. However, such numbering systems exhibit a limited dynamic range and poor accuracy when bit-widths are aggressively reduced below eight bits, resulting in severe rounding and overflow errors that distort spike timing and neuronal dynamics.

The POSIT number system [1] has emerged as an attractive alternative, offering a tapered precision that provides offers higher accuracy for small magnitudes while retaining low hardware complexity. In this work, we employ a 6-bit POSIT representation for the computation of synaptic currents and membrane potentials in SNNs. While this representation substantially reduces area and energy, it introduces new challenges related to accumulation instability. Such errors accumulate over time, leading to drift in neuronal states and degradation of inference accuracy.

To mitigate this problem, we integrate Kahan's compensated summation algorithm [5] into the accumulation stage of the SNN. By maintaining an auxiliary correction term, Kahan summation restores the low-order bits lost during addition, preserving numerical stability without increasing bit-width or hardware complexity. Experimental results show that our method achieves substantial improvements in both accuracy and stability, with negligible overhead.

The main contributions of this paper are summarized as follows:

- Introduction of a 6-bit POSIT arithmetic framework for SNN computation;
- Identification and quantitative analysis of accumulation errors under extreme quantization;
- Integration of Kahan summation for precision compensation in low-precision POSIT arithmetic;

The remainder of this paper is organized as follows. Section II reviews related work and background on SNN-arithmetic and the POSIT number system. Section III presents the proposed POSIT-based computation model and the integration of Kahan summation. Section IV reports experimental results and analysis. Finally, Section V concludes the paper and outlines future research directions.

## II. BACKGROUND AND RELATED WORK

Spiking neural networks offer an event-driven alternative to traditional deep learning models, enabling energy-efficient

978-1-5386-2585-9/25/$31.00 ©2025 IEEE

inference on neuromorphic hardware. However, their performance depends critically on numeric representation. The high dynamic range of neuronal states and the repeated accumulation of small synaptic inputs make low-bit arithmetic prone to precision loss, which can destabilize spike generation. Consequently, a large body of work has focused on quantization, reduced precision computation, and alternative number systems to achieve an optimal trade-off between accuracy and efficiency.

### A. Reduced-Precision Computing in SNNs

Existing SNN compression methods include pruning, knowledge distillation, and quantization [5]. Among these, quantization is particularly appealing for hardware deployment, as it directly reduces memory bandwidth and energy consumption. Most quantized SNN approaches constrain synaptic weights or membrane potentials to 2–8 bits [3], [6], achieving substantial power reduction but at the cost of accuracy degradation. Binary and ternary SNNs, for instance, often suffer from severe information loss and unstable firing dynamics during inference [7], [8].

Recent advances such as Q-SNN [8] and Fast-SNN [7] have improved quantization-aware training by integrating weight-spike co-optimization and surrogate gradient methods, However, these techniques still depend on floating-point arithmetic during training or accumulate quantization errors during inference. In general, reduced-precision training for SNNs requires extensive hyperparameter tuning to balance convergence, neuron thresholds, and firing rates [6].

### B. POSIT Arithmetic for Neural Network

The POSIT number system, first introduced by Gustafson and Yonemoto [1], represents real numbers using four variable-length fields: sign, regime, exponent, and fraction. This structure provides tapered precision, allowing high accuracy for small magnitudes while maintaining a wide representable range. Compared to IEEE-754 floating-point, POSIT eliminates redundant zero and infinity representations, reduces rounding modes, and simplifies hardware implementation [9]. A POSIT configuration is denoted as $\langle n, es \rangle$ where $n$ is total bit-width and *es* defines the maximum exponent size. For instance, a 6-bit POSIT with $es = 2$ can represent values in approximately [-65536, +65536] with superior resolution near unity.

Since its introduction, POSIT arithmetic has gained traction in deep learning as a hardware-friendly alternative to floating-point computation. Several frameworks, such as Deep PeNSieve [10], have demonstrated the potential of POSIT for both inference and training. These studies show that 8-bit POSIT inference can achieve accuracy comparable to 16-bit floating-point while reducing energy consumption. However, most existing implementations either rely on software emulation or hardware generators such as FloPoCo [11], and very few explore the ultra-low-bit regime ($\leqslant$6 bits) where hardware benefits are maximal but numerical instability becomes critical.

### C. Limitations and Motivations

Despite its advantages, POSIT arithmetic exhibits instability when operating at extremely low bit-widths. Accumulating values that differ significantly in magnitude can lead to catastrophic cancellation, where smaller operands are rounded away (e.g., 10000 + 1 → 10000). This issue mirrors precision loss in floating-point systems but is amplified in low-bit POSIT computations used for SNN integration. Prior studies have primarily focused on higher-bit (8 – 16 bit) POSIT applications in CNNs or DNNs, with limited attention to event-driven SNN architectures that rely on frequent accumulation and threshold updates.

### D. Kahan-Compensated Summation

To mitigate rounding errors that arise during accumulation, Kahan's compensated summation algorithm introduces a correction variable that tracks the small errors lost in each addition. Instead of directly summing the inputs, the algorithm maintains a compensation term $c$ that preserves the bits lost to rounding. This allows the final accumulated result to retain higher numerical fidelity without increasing bit width.

The pseudocode for the Kahan summation algorithm is Algorithm 1:

**ALGORITHM 1: KAHAN SUMMATION**

```
    Input: Array of values x[1...n]
    Output: Accurate accumulated sum, S
1   S ← 0        // running sum
2   c ← 0        // compensation term
3   for i = 1 to n do
4       y ← x[i] - c
5       t ← S + y
6       c ← (t - S) - y
7       S ← t
8   end for
9   Return S
```

At each iteration, the algorithm corrects the lost precision by subtracting the compensation term before the addition and recomputing it afterward. This ensures that small contributions often lost in low-bit POSIT arithmetic are progressively restored in later steps. When embedded within SNN synaptic integration, the algorithm preserves the fine-grained effect of weaker synaptic currents, leading to more stable membrane potential evolution and spike generation.

### E. Six-Bit POSIT Representation

The POSIT format represents a real number $x$ using a sign bit, a variable-length regime field, an optional exponent field, and a fraction field. The value of a POSIT number is computed as:

$$x = (-1)^s \times useed^r \times 2^e \times (1 + f) \qquad (1)$$

where $s$ is the sign, $r$ the regime value, $e$ the exponent, and $f$ the fractional part. $useed$ is defined as $2^{2^{es}}$, where $es$ is the number of exponent bits. The regime value $k$ depends on the number of identical bits at the start of the regime field, denoted as $r$. If the regime starts with a sequence of 1s followed by a 0, then $k$ equals $r$ and if it starts with a sequence of 0s followed by a 1, then $k$ equals $-(r + 1)$**,** Altogether these parts define the final value of a POSIT number

In this work, we employ a 6-bit POSIT configuration with $es = 2$, which provides a compact range of approximately [-65536, +65536] while maintaining higher relative precision for values near zero. Compared to a 6-bit fixed-point format, the POSIT encoding achieves better dynamic scaling, avoiding saturation in most membrane potential updates. All computations, including synaptic weights and neuron potentials,

are stored in this 6-bit format to reduce memory footprint and power consumption.

### F. Accumulation and Numerical Instability

In SNNs, each neuron integrates weighted spike inputs as:

$$V_{mem}(t+1) = V_{mem}(t) + \sum_i \omega_i \ . s_i(t) \qquad (2)$$

where $\omega_i$ is the synaptic weight and $s_i(t)$ the binary spike input. Under 6-bit precision, additions of small and large operands (e.g., $\omega_i s_i(t) \ll V_{mem}(t)$ ) cause rounding errors, leading to cumulative drift. Since the SNN's firing threshold depends on accurate potential tracking, even minor accumulation errors can shift spike timing and degrade inference accuracy.

Therefore, this work bridges these two directions, quantized SNNs and low-bit POSIT arithmetic, by introducing a 6-bit POSIT SNN framework enhanced with Kahan's compensated summation algorithm. This combination mitigates rounding-induced drift while preserving the compactness and energy efficiency of low-bit hardware, providing a practical and numerically stable solution for neuromorphic processors.

## III. Proposed Approach

This section introduces the proposed low-bit arithmetic framework for event-driven SNN computations. The design integrates a 6-bit POSIT number system with Kahan's compensated summation algorithm to improve numerical stability during synaptic accumulation. Figure 1 illustrates the general structure of our proposed model, showcasing how these components work together.

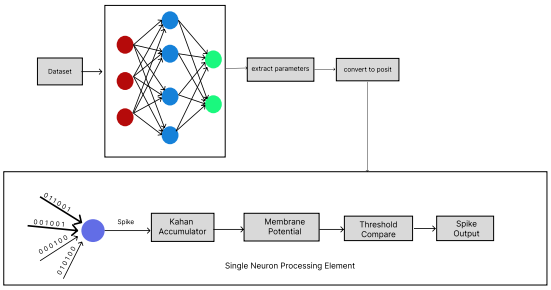


*Figure. 1. General structure of the proposed 6-bit POSIT + Kahan summation framework for event-driven SNN.*

### A. 6-bit POSIT Conversion Algorithm

The conversion of floating-point numbers to the 6-bit POSIT format is central to the proposed low-precision arithmetic in SNNs. The conversion function maps a floating-point input $x$ to a POSIT<6,2> representation consisting of a sign bit, a variable-length regime field, a 2-bit exponent, and the remaining fraction bits. Here, $es = 2$, which yields $useed = 16$ and determines the dynamic range of the number system. Special values (such as zero and NaN) and the sign are handled first, after which $|x|$ is normalized and the regime, exponent, and fraction fields are derived. The overall procedure is summarized in Algorithm 2.

**Algorithm 2: 6-bit POSIT Conversion**

***Input**: a floating-point number x*
***Output**: y = a 6-bit POSIT string POSIT <6, 2>*
***Function**: decimal_to_posit6_2(x)*
  ***if** x == 0 **or** x is NaN **then**,*
    ***return** the corresponding special POSIT representation*
  ***end if***
  *Determine the sign of **x***
  *Compute the regime by adjusting **x** with **useed** = $2^{2^{es}}$*
  *Calculate the exponent (2 bits) by normalizing **x***
  *Generate the regime bits based on the computed **k***
  *Calculate the fractional part of **x** and store the corresponding bits*
  *Combine regime, exponent, and fraction, and store the result in y*
  ***if** y is negative **then***
    *apply two's complement to **y***
  ***end if***
***Return** y*

## IV. Experimental Results

In this section, we present the experimental results of our proposed 6-bit POSIT + Kahan summation framework for Spiking Neural Networks, comparing it with alternative representations including 8-bit POSIT, 8-bit Fixed-Point, and 32-bit Floating-Point. The models were evaluated on the MNIST dataset, and performance was assessed in terms of accuracy, hardware resource utilization (LUT and FF), and energy efficiency.

### A. Experimental Setup

The experimental setup involved training and testing SNNs on the MNIST dataset, consisting of 28x28 pixel images. We implemented the models using Vitis HLS (High-Level Synthesis) and deployed them on a Zynq 7020 FPGA. Each configuration was synthesized and optimized for hardware using the Vitis HLS tool. The performance metrics considered include:

- LUT (Look-Up Tables): A measure of FPGA logic resource usage.
- FF (Flip-Flops): A measure of distributed memory resource utilization.

### B. Result Comparison

Table I presents the comparison between the proposed method and other numeric formats. The baseline 8-bit fixed-point implementation achieves only 66.9% accuracy, primarily due to rounding and saturation effects, although it requires the least hardware resources. In contrast, the proposed 6-bit POSIT configuration attains 93.2% accuracy, representing a 26.3% improvement over the fixed-point design while maintaining a reasonable hardware resource cost. The 8-bit POSIT format achieves a slightly higher accuracy of 95.1%, at the expense of increased LUT and FF utilization. Compared to the 32-bit floating-point baseline (97.6%), the 6-bit POSIT achieves competitive accuracy with significantly reduced hardware overhead. Furthermore, the integration of Kahan's compensated summation effectively mitigates accumulation drift, enabling near full-precision behavior in low-bit POSIT arithmetic.

*TABLE I. Performance Comparison of Numeric Formats*

| Format | Accuracy (%) | # LUT | # FF |
|---|---|---|---|
| **6-bit POSIT (Proposed)** | **93.2** | **31955** | **7110** |
| **8-bit POSIT** | 95.1 | 35273 | 9380 |
| **8-bit Fixed-Point** | 66.9 | 21174 | 8300 |
| **32-bit Floating Point** | 97.6 | 43918 | 15975 |

It is worth noting that, unlike conventional quantization-based designs, which often perform intermediate computations using higher bit-widths and only quantize the final outputs, the proposed method maintains uniform low-bit precision throughout all computations. This fully low-precision approach simplifies hardware implementation and ensures consistent numeric behavior across all processing stages, while still delivering competitive accuracy.

### C. Discussion

The proposed 6-bit POSIT representation combined with Kahan summation achieves an effective tradeoff between numerical accuracy and hardware resource cost. Although 8-bit POSIT and 32-bit floating-point implementations provide higher accuracy, they incur substantially greater FPGA resource utilization, rendering them less suitable for resource-constrained neuromorphic processors. In contrast, the 6-bit POSIT configuration attains accuracy comparable to the 8-bit POSIT model and significantly surpasses the fixed-point representation, while requiring considerably reduced hardware requirements. Furthermore, the use of Kahan summation effectively reduces rounding errors and enhances the numerical stability of the 6-bit POSIT-based accumulation, thereby maintaining high inference accuracy with reduced energy consumption.Hardware Efficiency and Energy Considerations

The results indicate that the 6-bit POSIT configuration is the most energy-efficient configuration, offering the best trade-off between accuracy and hardware resource usage. In contrast, both 8-bit POSIT and 32-bit floating-point models, although more accurate, incur a substantial overhead in terms of power and logic resource utilization, making them less viable for low-power neuromorphic hardware.

Figure 2 compares the accuracy and resource usage (LUT and filp-flops) across different numeric formats. The 6-bit POSIT (proposed) model is shown to achieve competitive accuracy while utilizing significantly fewer FPGA resources compared to the other formats.

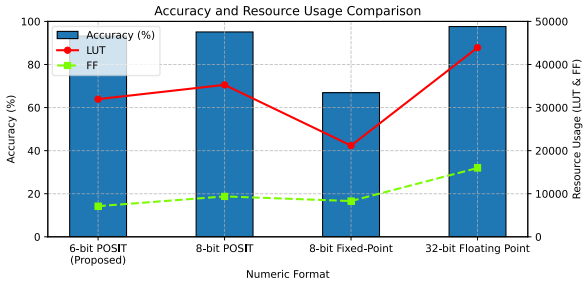


*Figure 2. Accuracy and resource usage comparison of different numeric formats. The 6-bit POSIT (proposed) representation achieves competitive accuracy with the lowest resource consumption*

To position the proposed 6-bit POSIT-based SNN, we compare it with representative neuromorphic and FPGA-based SNN accelerators on MNIST [12]–[18]. As summarized in Table 2, the proposed design achieves 93.2% accuracy using an ultra-compact 6-bit posit representation on FPGA, which is higher than several low-cost implementations such as Minitaur [15], Spiker [17], and the low-cost spiking ELM [18], while employing fewer bits than high-accuracy designs on SpiNNaker, TrueNorth, Darwin NPU, and ZC706-based SNNs [12]–[16]. This comparison demonstrates a favorable accuracy–precision trade-off and confirms that aggressive 6-bit posit quantization remains viable for edge neuromorphic hardware.

*TABLE II. Performance Comparison of Numeric Formats*

| Work | Platform | Precision (bits) | Network / Training | Accuracy (%) |
|---|---|---|---|---|
| **This work** | FPGA | 6-bit POSIT | SNN, offline | 93.2 |
| **Stromatias et al. [12]** | ASIC | 16 | Spike-based MLP | 95.0 |
| **Esser et al. [13]** | ASIC | Low-precision | Trained SNN | 95.0 |
| **Darwin NPU [14]** | ASIC | 8–16 | SNN co-processor | 93.8 |
| **Minitaur [15]** | FPGA | 16 | Event-driven SNN | 92.0 |
| **Han et al. [16]** | FPGA | 16 | Converted SNNs | 97.06 |
| **Spiker [17]** | FPGA | 16 / 5 | LIF SNN | 73.96 |
| **He et al. [18]** | FPGA | Low-precision fixed-point | Spiking ELM | 93.0 |

## V. Conclusion

In this paper, we proposed a 6-bit POSIT arithmetic framework for event-driven SNNs, enhanced with Kahan's compensated summation algorithm to improve numerical stability. Our results demonstrate that the 6-bit POSIT model offers a strong balance between accuracy and resource efficiency, achieving 93.2% accuracy while using significantly fewer FPGA resources compared to other numeric formats, such as 8-bit POSIT and 32-bit floating-point.

While 8-bit POSIT and 32-bit floating-point models provide slightly higher accuracy, they incur substantial overhead in terms of LUTs and flip-flops. The 6-bit POSIT + Kahan approach, however, mitigates precision loss during synaptic accumulation and maintains nearly full-precision performance with minimal resource utilization, making it a promising solution for low-power hardware.

Future work can focus on adaptive precision in POSIT encoding and the integration of this framework into on-chip learning systems. Additionally, we plan to explore ASIC implementations to further optimize the trade-off between accuracy and hardware resources.

## References


[1] J. L. Gustafson and I. T. Yonemoto, "Beating floating point at its own game: Posit arithmetic," *Supercomputing Frontiers and Innovations*, vol. 4, no. 2, pp. 71–86, 2017.

[2] B. Han and K. Roy, "Deep spiking neural network: Energy efficiency through time-based coding," in *Proc. European Conf. on Computer Vision (ECCV)*, Cham, Switzerland: Springer International Publishing, 2020, pp. 388–404.

[3] H. Xie, G. Yang, and W. Gao, "Toward efficient deep spiking neuron networks: A survey on compression," in *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Singapore: Springer Nature Singapore, July 2024, pp. 18–31.

[4] K. Yamazaki, V. K. Vo-Ho, D. Bulsara, and N. Le, "Spiking neural networks and their applications: A review," *Brain Sciences*, vol. 12, no. 7, p. 863, 2022.

[5] W. Kahan, “A survey of error analysis,” Univ. of California, Berkeley, CA, Tech. Rep. TR4, 1971.

[6] F. Cortney, “A survey on network quantization techniques for deep neural network compression,” *Authorea Preprints*, 2024. [Online]. Available: https://www.authorea.com

[7] C. Li, L. Ma, and S. Furber, “Quantization framework for fast spiking neural networks,” *Frontiers in Neuroscience*, vol. 16, 2022.

[8] W. Wei, Y. Liang, A. Belatreche, Y. Xiao, H. Cao, Z. Ren, G. Wang, M. Zhang, and Y. Yang, “Q-SNNs: Quantized spiking neural networks,” in *Proc. 32nd ACM Int. Conf. on Multimedia (ACM MM)*, Oct. 2024, pp. 8441–8450.

[9] N. Nakasato, Y. Murakami, F. Kono, and M. Nakata, “Evaluation of posit arithmetic with accelerators,” *in Proc. Int. Conf. on High Performance Computing in Asia-Pacific Region (HPC Asia)*, Jan. 2024, pp. 62–72.

[10] R. Murillo, A. A. Del Barrio, and G. Botella, “Deep PeNSieve: A deep learning framework based on the posit number system,” *Digital Signal Processing*, vol. 102, p. 102762, 2020.

[11] F. De Dinechin and B. Pasca, “Designing custom arithmetic data paths with FloPoCo,” *IEEE Design & Test of Computers*, vol. 28, no. 4, pp. 18–27, 2011.

[12] E. Stromatias, D. Neil, F. Galluppi, M. Pfeiffer, S.-C. Liu, and S. B. Furber, “Scalable energy-efficient, low-latency implementations of spiking deep belief networks on SpiNNaker,” *in Proc. Int. Joint Conf. on Neural Networks (IJCNN)*, Killarney, Ireland, Jul. 2015, pp. 1–8.

[13] S. K. Esser, R. Appuswamy, P. Merolla, J. V. Arthur, and D. S. Modha, “Backpropagation for energy-efficient neuromorphic computing,” in *Adv. Neural Inf. Process. Syst.* (NIPS), vol. 28, 2015, pp. 1117–1125.

[14] J. Shen, D. Ma, Z. Gu, M. Zhang, X. Zhu, X. Xu, Q. Xu, Y. Shen, and G. Pan, “Darwin: a neuromorphic hardware co-processor based on spiking neural networks,” *Sci. China Inf. Sci.*, vol. 59, no. 2, Art. no. 022401, Feb. 2016.

[15] D. Neil and S.-C. Liu, “Minitaur, an event-driven FPGA-based spiking network accelerator,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 22, no. 12, pp. 2621–2628, Dec. 2014.

[16] J. Han, Z. Li, W. Zheng, Y. Zhang, and J. Li, “Hardware implementation of spiking neural networks on FPGA,” *Tsinghua Sci. Technol.*, vol. 25, no. 4, pp. 479–486, Aug. 2020.

[17] A. Carpegna, A. Savino, and S. Di Carlo, “Spiker: an FPGA-optimized hardware accelerator for spiking neural networks,” in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Limassol, Cyprus, Jul. 2022, pp. 14–19.

[18] Z. He, C. Shi, T. Wang, Y. Wang, M. Tian, X. Zhou, P. Li, L. Liu, N. Wu, and G. Luo, “A low-cost FPGA implementation of spiking extreme learning machine with on-chip reward-modulated STDP learning,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 3, pp. 1657–1661, Mar. 2022.