Efficient yet Accurate End-to-End SC Accelerator Design

Meng Li^{213*}, Yixuan Hu¹, Tengyu Zhang¹, Renjie Wei¹, Yawen Zhang⁴, Ru Huang¹³⁴ and Runsheng Wang^{134*}

¹School of Integrated Circuits & ²Institute for Artificial Intelligence, Peking University, China

³Beijing Advanced Innovation Center for Integrated Circuits, Beijing, China

⁴Institute of Electronic Design Automation, Peking University, Wuxi, China

Abstract-Providing end-to-end stochastic computing (SC) neural network acceleration for state-of-the-art (SOTA) models has become an increasingly challenging task, requiring the pursuit of accuracy while maintaining efficiency. It also necessitates flexible support for different types and sizes of operations in models by end-to-end SC circuits. In this paper, we summarize our recent research on end-to-end SC neural network acceleration. We introduce an accurate end-to-end SC accelerator based on deterministic coding and sorting network. In addition, we propose an SC-friendly model that combines low-precision data paths with high-precision residuals. We introduce approximate computing techniques to optimize SC nonlinear adders and provide some new SC designs for arithmetic operations required by SOTA models. Overall, our approach allows for further significant improvements in circuit efficiency, flexibility, and compatibility through circuit design and model co-optimization. The results demonstrate that the proposed endto-end SC architecture achieves accurate and efficient neural network acceleration while flexibly accommodating model requirements, showcasing the potential of SC in neural network acceleration.

I. INTRODUCTION

Stochastic computing (SC) has emerged as a promising alternative to traditional binary computing, offering simplified arithmetic operations and improved error resilience [1]–[5]. Both hybrid and end-to-end SC-based neural accelerators have been proposed [1]–[5]. While hybrid accelerators involve back-and-forth conversion between binary and SC representations, leading to high power consumption and area overhead, end-to-end SC-based accelerators demonstrate superior power, area efficiency, and fault tolerance [3]–[5]. In this context, our research aims to further enhance the capabilities of end-to-end SC-based accelerators.

Existing SC-based accelerators primarily focus on multiplication, accumulation, and activation functions in convolutional networks [6]–[9]. However, these approaches have limitations. FSM-based activation modules suffer from accuracy issues, particularly for ReLU with larger accumulation widths (Figure 1). Furthermore, there exists a trade-off between inference efficiency and accuracy (Figure 2), where high precision computing enhances accuracy but exponentially increases costs, while low precision computing compromises accuracy. Additionally, there is a lack of research on SC circuits supporting functions like batch normalization (BN), residual connections, gaussian error linear unit (GELU), and softmax for state-of-the-art (SOTA) models.

Therefore, in this paper, we will summarize our recent efforts on end-to-end SC-based NN accelerators that address these limitations to meet the requirements in terms of accuracy, efficiency, flexibility, and compatibility, as shown in Table I.

II. ACCURATE END-TO-END SC ACCELERATOR BASED ON DETERMINISTIC THERMOMETER CODING

In this section, we introduce deterministic coding called thermometer coding and the corresponding end-to-end SC accelerator design. The proposed design achieves exact end-to-end SC NN acceleration.

This work was supported in part by the National Key R&D Program of China (2020YFB2205502), NSFC (62125401) and the 111 Project (B18001). *Corresponding author: {meng.li, r.wang}@pku.edu.cn

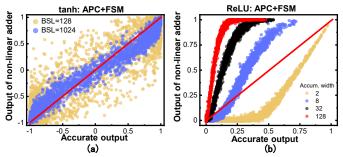


Fig. 1. FSM-based design to implement (a) tanh and (b) ReLU. Ideally, the circuit output is the same as the exact output, marked by the red line.

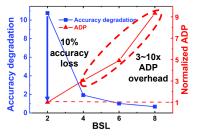


Fig. 2. The trade-off between inference accuracy and efficiency (measured by area-delay product, i.e., ADP). Here, we fix the weight BSL to 2-bit and sweep the activation BSL.

TABLE I COMPARISON OF DIFFERENT END-TO-END SC ACCELERATORS.

Design	Accuracy	Efficiency	Flexibility	*Compatibility
FSM-based [6]-[9]	Low	Low	Limited for large Conv	Basic CNNs
Ours [3]-[5]	High	Low	Limited for variable Conv	Basic CNNs
Ours [10], [11]	High	High	Flexible	DNNs
Ours [12]	High	High	Flexible	DNNs+ViT

*Basic CNNs contain convolution and ReLU. DNNs further require residual connection and BN. And transformer models further require GeLU and softmax.

A. Motivation

We refer to the accumulation and activation module as the SC non-linear adder. Typical SC Non-linear adders employ stochastic coding with FSM to implement different activation functions [6]–[9]. FSM-based designs serially process stochastic bitstream inputs, which results in inaccurate outputs (Figure 1) that do not utilize all of the information in the inputs and have random fluctuations in the inputs themselves. Thus, very long bitstreams, e.g., 1024 bits, are used for accuracy and lead to an unacceptable latency, which severely affects the hardware efficiency.

B. Accurate End-to-End SC Acceleration with Sorting Network

In our work, we employ the deterministic thermometer coding scheme (Table II) and the corresponding accurate SC circuit designs to achieve accurate end-to-end SC acceleration. With thermometer coding, all the 1s appear at the beginning of the bitstream and

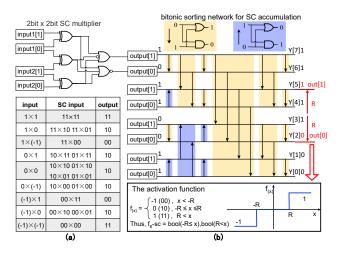


Fig. 3. (a) The truth table and circuit of ternary SC multiplier. (b) The BSN and the selective interconnect system for accumulation and activation function.

TABLE II
THE CORRESPONDING BINARY PRECISION AND THE REPRESENTED RANGE
FOR THERMOMETER CODING OF DIFFERENT BSL.

BSL	Binary Precision	Range	Thermometer Coding
2	-	-1, 0, 1	00, 10, 11
4	2	-2, -1, 0, 1, 2	0000, 1000, 1100, 1110, 1111
8	3	-4, -3 · · · 3, 4	00000000, 100000000 · · · 111111110, 11111111
16	4	-8, -7 · · · 7, 8	00000000000000000, 10000000000000000 · · · 1111111111

a value x is represented with a L-bit sequence as $x=\alpha x_q=\alpha(\sum_{i=0}^{L-1}x[i]-\frac{L}{2})$, where $x_q=\sum_{i=0}^{L-1}x[i]-\frac{L}{2}$ is the quantized value of range $[-\frac{L}{2},\frac{L}{2}]$ and α is a scaling factor obtained by training.

Deterministic coding, in contrast to stochastic coding, achieves hardware-efficient and accurate computations with shorter bitstreams. By employing a 2-bit ternary bitstream, we can realize multiplication with only 5 gates using a deterministic multiplier (Figure 3(a)).

To achieve accurate accumulation and activation functions simultaneously, we employ the bitonic sorting network (BSN). BSN is a parallel sorting network that sorts inputs in thermometer coding, ensuring the output is also in thermometer coding. The sorting process, performed by comparators constructed with AND and OR gates, follows Batcher's bitonic sorting algorithm [13] (Figure 3(b)). The number of 1's in the sorted bitstream output from BSN corresponds to the sum of 1's in all input bitstreams, effectively representing the accumulation result.

By sorting all the bits, the inputs and outputs of the selective interconnect (SI) [14] are deterministic. Therefore, when the SI selects different bits from the BSN directly as outputs based on the selection signals, a deterministic input-output correspondence is generated and different activation functions are realized. The example in Figure 3(b) implements the two-step activation function shown at the bottom when the SI selects the 3rd and 6th bits of the BSN as outputs. We refer interested readers to [3], [4] for more details.

C. Experimental Results

We prototype the proposed SC accelerator with a 28-nm CMOS process. The chip's measured current consumption and energy efficiency in Figure 4 show a peak of 198.9 TOPS/W at 200 MHz and 650 mV. Compared to state-of-the-art binary-based NN processors [15]–[19], the fabricated SC-based NN processor achieves an average

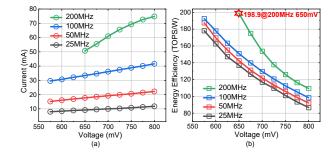


Fig. 4. (a) Current and (b) energy efficiency versus supply voltage at different working frequencies.

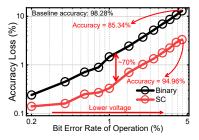


Fig. 5. Accuracy loss of the conventional binary design and proposed SC design versus bit error rate, at the soft accuracy of 98.28%.

energy efficiency improvement of $10.75 \times (1.16 \times \sim 17.30 \times)$. And the area efficiency improves by $4.20 \times (2.09 \times \sim 6.76 \times)$. We also compare the accuracy under varying bit error rates (BER) using a ternary neural network that achieves 98.28% accuracy on the MNIST dataset, as shown in Figure 5. The proposed SC design demonstrates significant fault tolerance, as the average reduction of accuracy loss by 70%. It is the first silicon-proven end-to-end SC accelerator, to the best of the authors' knowledge.

III. ACCURATE YET EFFICIENT SC WITH HIGH PRECISION RESIDUAL FUSION

The SC accelerator above validated the effectiveness of deterministic thermometer coding and the corresponding SC design on the basic small model (MNIST). In this section, we propose SC-friendly models as well as new SC circuit blocks to support SOTA model requirements and greatly improve the accuracy of the SC accelerators.

A. Motivation

The SC TNN accelerator in Section II lacks support for batch normalization (BN) and residual connections, limiting its accuracy on complex datasets like CIFAR10 or CIFAR100. Increasing precision can enhance accuracy but compromises hardware efficiency. Figure 2 demonstrates that increasing BSL from 2 to 8 bits improves accuracy at the expense of a 3 to 10 times efficiency overhead. Accurate yet efficient SC acceleration is very challenging.

B. SC-Friendly Low Precision Network

To understand the origin of the accuracy degradation, we quantize the network weight and activation to low precision separately. Table III shows similar accuracy between low precision weight quantization and the floating point baseline, while 2b BSL activation quantization results in a 10% accuracy drop. Hence, low precision activation is the root cause of the accuracy loss due to its limited representation capacity. After quantization, the range of activations is

TABLE III
NETWORK ACCURACY COMPARISON OF DIFFERENT QUANTIZED
NETWORKS ON CIFAR 10.

Network	Weight/BSL	Act/BSL	Top-1 Accuracy (%)
baseline	FP	FP	94.27
weight quantized	2	FP	93.98
activation quantized	FP	2	84.18
fully quantized	2	2	83.51

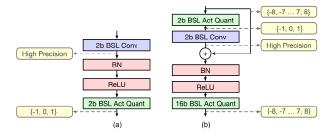


Fig. 6. High precision residual helps to achieve better representation capability.

reduced to $\{-1,0,+1\}$ for 2b BSL encoding, significantly reducing the number of possible configurations.

As a remedy, we add the high-precision activation input through residual connections to the result of the low-precision convolution (Figure 6). By increasing the activation range to $\{-8, -7, \ldots, 7, 8\}$, we enhance representation capacity to $17^{H \times W \times C}$. This significantly improves inference accuracy while maintaining efficiency by preserving energy-efficient convolution computation.

$$ReLU(BN(x)) = \begin{cases} \gamma(x-\beta) & x \ge \beta \\ 0 & x < \beta \end{cases}$$
 (1)

Besides the high precision residual, another remaining question is how to efficiently process BN. And $BN(x) = \gamma(x-\beta)$, where γ and β are trainable parameters. We propose to fuse BN with the ReLU activation function as Equation 1. Consequently, we achieve an SC-friendly low precision model with high precision residual fusion depicted in Figure 6(b).

C. End-to-End SC Accelerator with High Precision Residual

Compared to the proposed accelerator in Section II-B, the model in Figure 6(b) further requires the implementation of SC circuits for BN fusion and residual connection.

The above fused BN and ReLU function can be efficiently and accurately processed in SC, leveraging the selective interconnect described in Section II-B. Figure 7 demonstrates how different BN parameters affect the objective function of the SI.

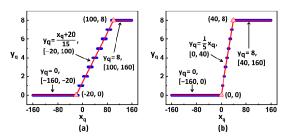


Fig. 7. BN-fused activation function with 16b BSL output. The blue dots are the outputs of the proposed design for the BN-fused ReLU (Equation 1).

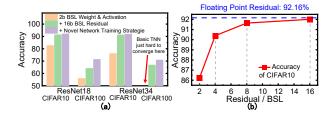


Fig. 8. (a) The proposed model optimization helps to achieve much better inference accuracy; (b) 16b BSL residual achieves 5.78% accuracy improvement, almost the same as floating point residual.

TABLE IV
INFERENCE EFFICIENCY AND ACCURACY COMPARISON.

W-A-R/BSL	Area (um²)	ADP (um²·us)	Accuracy (%)
2-2-2	4349.7	225.36	82.58
2-4-4	10683.3	687.47	92.35
2-2-16	4406.9	228.32	92.01

For the accumulation of residual and multiplication products, the different scaling factors α of residual and convolution results can lead to errors in the accumulation operation. The residual re-scaling block is proposed to align the α before accumulation. In the re-scaling block, we multiply or divide the residual by a factor of 2^N (where N is an integer). To multiply the residual by 2^N , we replicate it 2^N times in the buffer. For division by 2^N , we select 1 out of 2 bits of the residual per cycle and generate the final result after N cycles. To maintain a constant BSL for the residual, we append 8 bits of '11110000' (equal to 0) per division cycle.

D. Experimental Results

Figure 8 demonstrate significant improvement in network accuracy. With the high precision residual, network accuracy is improved significantly by 8.69% and 8.12% for low precision ResNet18 on CIFAR10 and CIFAR100, respectively. Combined with the novel training techniques, network accuracy can be improved in total by 9.43% and 15.42%. Compared to baseline accelerators, it achieves a 9.4% accuracy improvement with only a 1.3% efficiency overhead compared to the efficient baseline and achieves a 3× efficiency improvement with comparable accuracy to the accurate baseline design, as shown in Table IV. In this way, the proposed method achieves accurate yet efficient SC acceleration.

IV. FLEXIBLE AND EFFICIENT SC ACCELERATOR WITH APPROXIMATE SPATIAL-TEMPORAL BSN

In this section, we greatly improve the flexibility and hardware efficiency of the SC accelerator by compressing the BSN.

A. Motivation

BSN accumulates all the input in parallel through sorting, so as to generate an accurate output based on all the information input. However, it also forces the hardware cost to increase super linearly with the accumulation widths (Figure 9(a)). And the BSN has to support the largest accumulation widths among all layers. The large BSN, however, leads to very high hardware redundancy at shallow layers where the accumulation is always small (Figure 9(b)). This makes our previous design still inefficient for SOTA models.

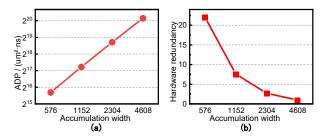


Fig. 9. The inefficiency of the BSN design: (a) BSN hardware cost increases significantly with the accumulation widths; (b) ADP overhead using a large BSN for small accumulation widths.

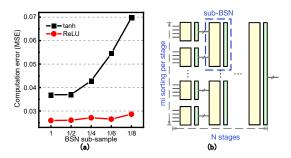


Fig. 10. (a) Reducing BSN output BSL has little effect on the accuracy of SI; (b) Parameterized BSN design space.

B. Approximate Spatial-Temporal Sorting Network

To address the inefficiency and inflexibility of BSN, we find a significant precision gap between the input and output of SI, as revealed in Figure 6(b), making the high precision SC input redundant. We reduce the BSN output BSL, resulting in a small accuracy loss for the tanh function and negligible impact on the ReLU function, as shown in Figure 10(a).

To further reduce hardware cost, we adopt a progressive sorting and sub-sampling approach for the BSN. Figure 10(b) presents a parameterized BSN design space that determines the location, number of sampling times, and method of sampling. The parameterized BSN consists of N stages and in the ith stage, there are m_i sub-BSN modules, each taking an input bitstream of l_i -bit BSL. Within each sub-BSN, there is a sub-sampling block that implements truncated quantization. It clips out c_i bits on each end of the BSN while sampling 1 bit every s_i bit from the remaining. Considering the input distribution resembles a Gaussian distribution with a small variance due to inputs from a large number of multipliers, significant clipping can be performed with negligible errors, as illustrated in Figure 11.

Thanks to the fact that the output BSL of the approximate BSN is much shorter than the input, we can further fold the accumulation temporally to achieve more flexibility. In this case, as shown in Figure 12, a large BSN is implemented by multi-cycle reuse of a single small BSN circuit. In the proposed spatial-temporal BSN architecture, the approximation level of BSN, i.e., the BSL of partial sums, and its corresponding reuse can be controlled through control signals. This allows for flexible handling of various accumulation widths with different approximate configurations.

C. Experimental Results

For the largest convolution in the ResNet18, the two proposed approximate BSN reduced the ADP of BSN by $2.8\times$ and $4.1\times$ compared to the baseline, as shown in Table V. When handling the four different sizes of convolutions in ResNet18, the spatial-temporal

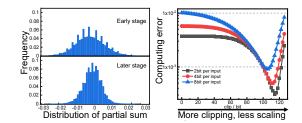


Fig. 11. Input distribution of the intermediate sub-sampling blocks in different stages of the BSN provides an opportunity to reduce the BSN via clipping.

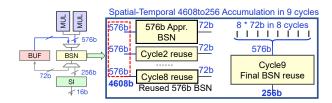


Fig. 12. Spatial-temporal BSN architecture with an example: a 576-bit BSN is reused over 9 cycles for 4608b accumulation.

 $TABLE\ V \\ Performance\ of\ different\ designs\ for\ a\ 3x3x512\ convolution.$

Design	Area (um²)	Delay (ns)	ADP (um2·ns)	MSE
Baseline BSN	2.95×10^{5}	4.33	1.26×10^{6}	-
Spatial Appr. BSN	1.32×10^{5}	3.36	4.55×10^{5}	3.79×10^{-7}
Spatial-Temporal Appr. BSN	8.18×10^{3}	1.92	3.06×10^{5} *	3.79×10^{-7}

*Spatial-temporal BSN considers 19× area to achieve the same throughput.

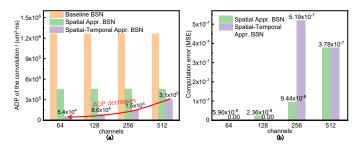


Fig. 13. (a) ADP and (b) MSE comparison on 4 size of layers in ResNet18.

BSN needs fewer cycles for smaller convolutions and achieved ADP reductions from $8.2\times$ to $23.3\times$ with negligible errors, as shown in Figure 13. On average, the spatial-temporal BSN reduces the $2.2\times$ area of datapath by reducing the average ADP of BSN by $8.5\times$. This shows that the proposed SC design is more flexible and efficient.

V. SUMMARY AND FUTURE WORK

In this paper, we review our recent works on end-to-end SC neural network acceleration. [4] implemented a parallel fully SC-based TNN processor using deterministic thermometer encoding and sorting networks on the MNIST, achieving energy efficiency of 198.9 TOPS/W. In addition, [10] propose SC-friendly models with high-precision residual fusion and corresponding SC circuits to greatly improve the network accuracy. [11] further proposed a more flexible and efficient spatial-temporal approximate BSN, enabling accurate, efficient, and flexible end-to-end SC acceleration. In future work, we explore SOTA transformer acceleration based on end-to-end stochastic computing, which has been submitted [12].

REFERENCES

- W. Romaszkan et al., "ACOUSTIC: Accelerating Convolutional Neural Networks through Or-Unipolar Skipped Stochastic Computing," in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020, pp. 768–773.
- [2] W. Romaszkan et al., "A 4.4–75-TOPS/W 14-nm Programmable, Performance- and Precision-Tunable All-Digital Stochastic Computing Neural Network Inference Accelerator," *IEEE Solid-State Circuits Letters*, vol. 5, pp. 206–209, 2022.
- [3] Y. Zhang et al., "When sorting network meets parallel bitstreams: A fault-tolerant parallel ternary neural network accelerator based on stochastic computing," in *Design, Automation & Test in Europe Con*ference & Exhibition (DATE). IEEE, 2020, pp. 1287–1290.
- [4] Y. Hu et al., "A 28-nm 198.9-TOPS/W Fault-Tolerant Stochastic Computing Neural Network Processor," IEEE Solid-State Circuits Letters, vol. 5, pp. 198–201, 2022.
- [5] Y. Zhang et al., "Accurate and Energy-Efficient Implementation of Non-Linear Adder in Parallel Stochastic Computing using Sorting Network," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.
- [6] K. Kim et al., "Dynamic energy-accuracy trade-off using stochastic computing in deep neural networks," in *Proceedings of the 53rd Annual Design Automation Conference*, 2016, pp. 1–6.
- [7] J. Li et al., "Towards acceleration of deep convolutional neural networks using stochastic computing," in 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC). IEEE, 2017, pp. 115–120.
- [8] Z. Li et al., "HEIF: Highly efficient stochastic computing-based inference framework for deep neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 8, pp. 1543–1556, 2018.
- [9] J. Li et al., "Hardware-driven nonlinear activation for stochastic computing based deep convolutional neural networks," in 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 1230–1236.
- [10] Y. Hu et al., "Accurate yet Efficient Stochastic Computing Neural Acceleration with High Precision Residual Fusion," in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2023.
- [11] Y. Hu et al., "Efficient Non-Linear Adder for Stochastic Computing with Approximate Spatial-Temporal Sorting Network," in ACM/IEEE Design Automation Conference (DAC), 2023.
- [12] Y. Hu *et al.*, "ASCEND: Accurate yet Efficient End-to-End Stochastic Computing Acceleration of Vision Transformer," in *submitted*.
- [13] K. E. Batcher, "Sorting networks and their applications," in *Proceedings of the April 30–May 2, 1968, spring joint computer conference*, 1968, pp. 307–314.
- [14] S. Mohajer et al., "Routing magic: Performing computations using routing networks and voting logic on unary encoded data," in Proceedings of the 2018 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2018, pp. 77–86.
- [15] J. Lee et al., "UNPU: A 50.6 TOPS/W unified deep neural network accelerator with 1b-to-16b fully-variable weight bit-precision," in 2018 IEEE International Solid-State Circuits Conference-(ISSCC). IEEE, 2018, pp. 218–220.
- [16] J. Song et al., "7.1 An 11.5 TOPS/W 1024-MAC butterfly structure dual-core sparsity-aware neural processing unit in 8nm flagship mobile SoC," in 2019 IEEE International Solid-State Circuits Conference-(ISSCC). IEEE, 2019, pp. 130–132.
- [17] C.-H. Lin *et al.*, "7.1 A 3.4-to-13.3 TOPS/W 3.6 TOPS dual-core deep-learning accelerator for versatile AI applications in 7nm 5G smartphone SoC," in *2020 ieee international solid-state circuits conference-(isscc)*. IEEE, 2020, pp. 134–136.
- [18] F. Tu et al., "Evolver: A deep learning processor with on-device quantization-voltage-frequency tuning," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 2, pp. 658–673, 2020.
- [19] H. Mo et al., "9.2 A 28nm 12.1 TOPS/W dual-mode CNN processor using effective-weight-based convolution and error-compensation-based prediction," in 2021 IEEE International Solid-State Circuits Conference (ISSCC), vol. 64. IEEE, 2021, pp. 146–148.