A 28nm 198.9 TOPS/W Fault-Tolerant Stochastic Computing Neural Network Processor

Yixuan Hu, Yawen Zhang, Runsheng Wang, *Member, IEEE*, Zuodong Zhang, *Graduate Student Member, IEEE*, Jiahao Song, *Graduate Student Member, IEEE*, Xiyuan Tang, *Member, IEEE*, Weikang Qian, *Senior Member, IEEE,* Yanzhi Wang, *Member, IEEE,* Yuan Wang, *Member, IEEE*, Ru Huang, *Fellow, IEEE*

***Abstract*—Previous energy-efficient neural network (NN) processors suffer from bit errors when operating at lower voltages for further power reduction. Stochastic computing (SC) shows great potential due to its low hardware cost and high fault tolerance. Conventionally, limited by the long latency of bitstreams, SC-based NN accelerators adopt a hybrid stochastic-binary architecture, sacrificing fault tolerance and hardware efficiency. This paper proposes a fully SC architecture that maximizes fault tolerance while offering excellent energy and area efficiency. The fabricated 28nm prototype is the *first* silicon-proven SC-based NN processor, realizing an energy efficiency of 198.9 TOPS/W and an area efficiency of 2630 GOPS/mm^2 with an accuracy loss reduction of 70%.**

***Index Terms*—stochastic computing (SC), thermometer-coding bitstream, low voltage, fault tolerance, bitonic sorting network.**

I. INTRODUCTION

The development of neural networks (NN) leads to complex hardware implementations and higher computational requirements. However, the process fluctuation increases with the CMOS device shrinking, resulting in the increased circuit error rate, which hinders further supply voltage and circuit power reduction [1]. A few techniques have been proposed in binary architecture to alleviate this issue. For example, the Razor system is adopted to reduce timing error [2][3]. Recently, stochastic computing (SC) has been proposed as a fault-tolerant alternative that uses the probability of '1's in a bitstream to represent a value [4], as shown in Fig. 1. In contrast to prior techniques, it mitigates the impact of bit error regardless of its causes. It is worth noting that the widely used serial SC multiplier is just an AND gate. Therefore, SC shows excellent potential for NN processors due to its low hardware cost of multiplier and good fault tolerance [5]-[8]. Those merits are especially attractive to low-precision quantization NNs that require simple hardware implementation and low fault rate, such as ternary neural networks (TNN). Recent research has greatly extended TNN's capability [9], making SC-based TNN processors promising solutions for

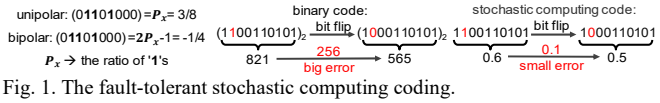


Fig. 1. The fault-tolerant stochastic computing coding.

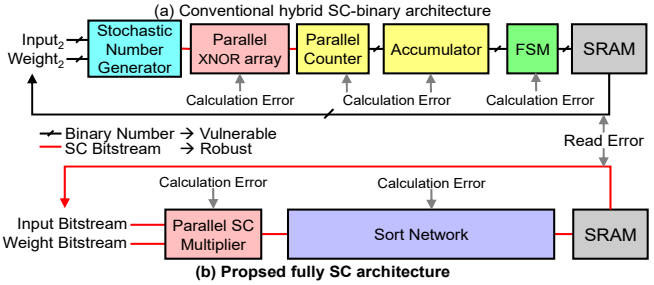


Fig. 2. The architectures of (a) hybrid SC-binary design and (b) the proposed fully SC-based NN processor.

emerging applications. However, the conventional SC-based accumulation and activation functions are inaccurate. To avoid accuracy degradation, the prior works adopt accumulator parallel counter (APC) or its approximation for addition [5][6] and finite state machines (FSMs) or linear approximation units for activation functions [7][8]. Those implementations require converting bitstream back into binary numbers to maintain the high accuracy of cascading operations in NNs, which increases the hardware cost and sacrifices the SC's fault tolerance. Besides, the long bitstreams are processed serially, severely degrading computation speed.

To address the issues above, this letter proposes a parallel fully SC-based TNN processor with three key contributions: 1) a parallel fully SC architecture that purely uses thermometer-coding bitstream (TCB), eliminating the conversion between SC bitstream and binary numbers; 2) a deterministic SC ternary multiplier without the randomness error of traditional SC; 3) using a bitonic sorting network (BSN), instead of APC and FSMs in traditional SC, to implement accurate and energy-efficient accumulator and activation functions. Combining these merits, the proposed SC-based TNN processor achieves 198.9 TOPS/W energy efficiency and 2630 GOPS/mm^2 area efficiency. Benefiting from the fault tolerance, the proposed SC

*This work was supported by National Key R&D Program of China (2020YFB2205502), and the 111 Project (B18001). The corresponding authors are *(Corresponding author: Runsheng Wang, Xiyuan Tang, Yuan Wang).*

Yixuan Hu, Yawen Zhang, Runsheng Wang, Zuodong Zhang, Jiahao Song, Yuan Wang, and Ru Huang are with the Key Laboratory of Microelectronic Devices and Circuits (MOE), School of Integrated Circuits, Peking University, Beijing 100871, China. (email: {wangyuan, r.wang}@pku.edu.cn)

Xiyuan Tang is with Institute for Artificial Intelligence and School of Integrated Circuits, Peking University, Beijing 100871, China. (email: xitang@pku.edu.cn).

Weikang Qian is with University of Michigan-Shanghai Jiao Tong University Joint Institute and MoE Key Laboratory of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai, China.

Yanzhi Wang is with the Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org

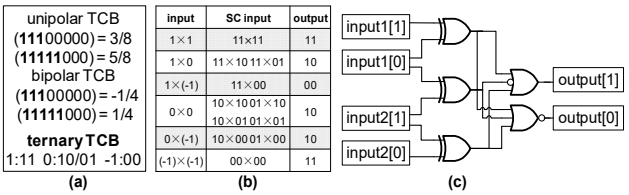


Fig. 3. (a) The forms of thermometer-coding bitstream (TCB). The (b) truth table and (c) circuit of ternary TCB multiplier.

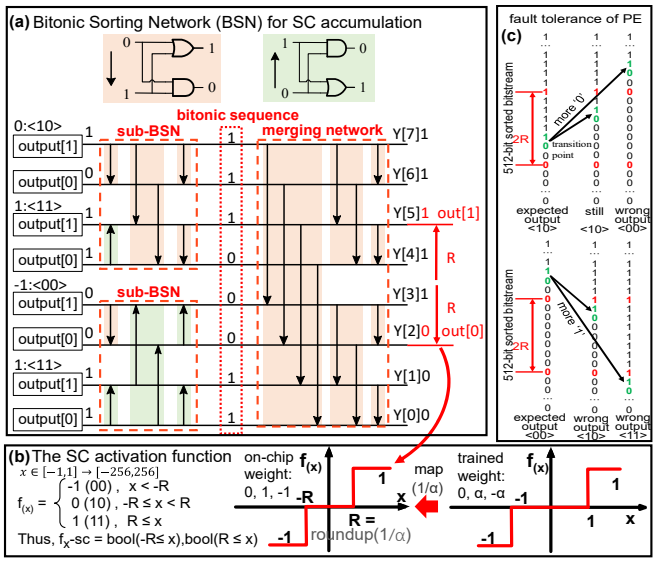


Fig. 4. (a) The BSN-based SC accumulation; (b) SC activation function based on the selective interconnect system; (c) The fault tolerance of PE.

architecture reduces accuracy loss by ~70% compared to the binary architecture under the same bit error rate (BER). This makes operations under low supply voltages feasible. To the best of the authors' knowledge, it is the first silicon-proven SC-based NN processor.

II. Proposed Architecture

When operating under low supply voltages, NN processors suffer from bit errors, such as processing element (PE) calculation errors and SRAM read errors. As shown in Fig. 1, the SC bitstream has good fault tolerance for bit errors, while the binary number is vulnerable due to the weighted bits. However, the hybrid SC-binary NN processor needs to convert the bitstream back to binary numbers, as shown in Fig. 2(a). Thus, SRAM read errors and calculation errors still appear in weighted binary numbers, weakening the fault tolerance. To address such challenges, the proposed SC-based architecture, as shown in Fig. 2(b), uses TCB exclusively for multiplication, accumulation, activation functions, and storage. Fig. 3(a) shows the forms of TCB. It only needs two bits to accurately represent the ternary value in TNN, with the same coding efficiency as binary code. Therefore, the cost of storing TCBs in SRAM is the same as storing conventional binary complement numbers.

Given the ternary TCB representation, we can obtain the implementation of each arithmetic unit in the PE. Fig. 3(b) is the truth table for the ideal ternary multiplication. Based on this, we derive a deterministic multiplier, requiring only five gates to realize multiplication in one cycle, as shown in Fig. 3(c).

Then, we employ BSN to implement accurate accumulation and activation functions simultaneously. BSN is a parallel

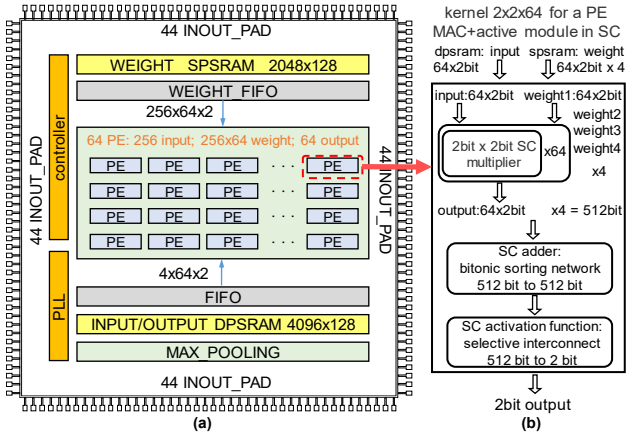


Fig. 5. (a) Overall architecture of the SC-based TNN processor; (b) The schematic and working mode of PEs.

sorting network designed to sort all inputs so that the output is also in TCB form, i.e., all the '1's before all the '0's. The arrow in the BSN is a comparator consisting of an AND gate and an OR gate, divided into two types: ascending and descending units. The arrows recursively construct the BSN according to Batcher's bitonic sorting algorithm [10]. In a 2N-bit BSN, as shown in Fig. 4(a), the two N-bit sub-BSNs first convert half of the input sequence into a monotonic sequence. The two monotonic sequences are joined at the end to form a bitonic sequence, which is then merged into the final 2N-bit TCB. Fig. 4(a) illustrates an 8-bit BSN with four inputs: 0, 1, -1, and 1. The two sub-BSNs produce a bitonic sequence of (11100011), and the output is an 8-bit bipolar TCB (11111000) with a value of 1/4. Although the hardware cost of the BSN-based parallel accumulator grows with input data, it can be relieved by processing the accumulation in steps.

The BSN does not change the number of '1's overall input TCBs. However, by sorting all the bits, a single output bit of the sorting network directly represents the comparison of the accumulation result with a certain value. For example, as shown in Fig. 4(a), if the sixth bit $Y_{[2]}$ of the sorting result is 0, it means that the output is less than 1/2 (11111100), i.e., $Y_{[2]} = \text{bool}(x \geqslant 1/2)$. Therefore, the selective interconnection system [11] can implement the desired activation function by setting the parameter *R* for different output bits. Here, we adopt a two-step function as the activation function for the ternary activation quantization, as shown in Fig. 4(b). In this case, the parameter *R* for the activation function in each convolutional layer is determined by the trained weights α. The *R* of the five on-chip layers in Fig. 6 are 3, 8, 8, 7, and 6, respectively.

In addition to implementing arithmetic logic functions, SC-based PEs are fault-tolerant. As the voltage decreases, PE calculation errors and SRAM read errors are more likely to occur, leading to the increased total BER. In this proposed SC architecture, the bit errors are primarily reflected as fluctuation in the number of '1's in the TCB after BSN. It can also be regarded as a forward or backward shift of the transition point between '1's and '0's, as shown in Fig. 4(c). If the shift of the transition point does not pass through the selected bit, the effect of the bit errors is eliminated. Only a large number of bit errors moving the transition point past both selected bits can cause an error of +2 or -2. This

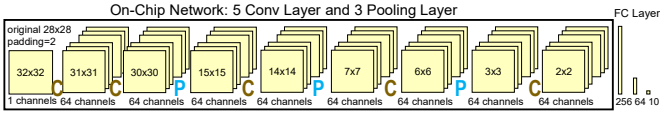


Fig. 6. The structure of proposed ternary neural network.

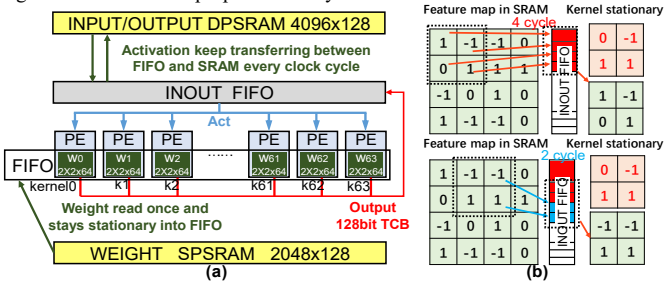


Fig. 7. (a) The weight stationary dataflow; (b) Convolution kernel calculation.

demonstrates the fault tolerance of PE.

The proposed PE uses deterministic multipliers, BSN, and the selective interconnect system to implement multiply-accumulate (MAC) and activation functions. The whole process is performed in parallel and completely adopts the SC bitstreams without conversion to binary numbers, which fully exploits the high fault tolerance of SC. Moreover, the PEs are completely accurate without any random fluctuations.

Fig. 5(a) presents the top-level architecture, consisting of PE array, weight single-port SRAM (SPSRAM), input/output dual-port SRAM (DPSRAM), FIFO, PLL, controller, and max-pooling unit. Bitstreams and all control signals are transmitted through 176 I/O PADs. Fig. 5(b) shows that each ternary PE consists of 256 SC multipliers and a 512-bit BSN. Considering that the size of BSN is a power of 2, extra energy will be wasted if the kernel and BSN do not match. For example, for a 3x3 kernel with 28 channels, the input is an unsorted 504-bit bitstream that can be sorted only after complementing with 8-bit 0's, degrading energy efficiency. Therefore, we use a convolutional kernel of even-size (2x2) with 64 channels in each layer to take full advantage of the 512-bit BSN. Each time, one PE reads an input vector and four weight vectors, where each vector comprises 64 ternary data, i.e., a 128-bit bitstream. Then the multipliers multiply four weight data and one input data on 64 channels. After obtaining the 256 ternary products, the 512-bit BSN performs accumulation and activation function operations to output a 2-bit ternary TCB. The results of 64 PEs, also a 128-bit bitstream, are then stored in DPSRAM arrays as the input data of the next layer. In this case, we realized this fully SC-based end-to-end TNN.

The TNN structure is modified from LeNet-5 [12], as shown in Fig. 6. All convolutional and pooling layers are implemented on-chip. All network parameters and image data are pre-stored in on-chip SRAM, and the 128 bits in SRAM correspond to the ternary data in 64 channels.

As shown in Fig. 7(a), we leverage the weight stationary data flow, which is universally applicable and widely used for network models [13]. The weight stationary dataflow is designed to minimize the weight access energy. The weight is read into the weight FIFO from SRAM at once and stays stationary for future access. The input/output FIFO fetches the data of all channels of one location in the DPSRAM every clock cycle. First, the input data of the first four cycles are sent to the

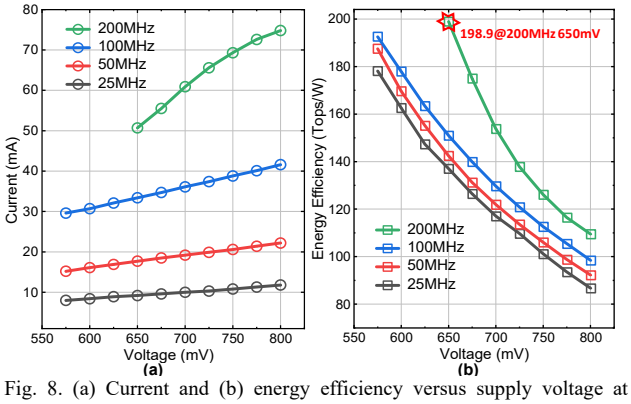


Fig. 8. (a) Current and (b) energy efficiency versus supply voltage at different working frequencies.

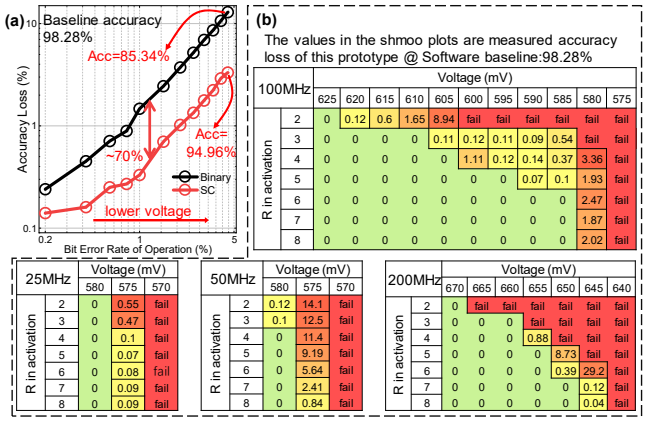


Fig.9. (a) Accuracy loss of the conventional binary design and proposed SC design versus bit error rate (BER), at the soft accuracy of 98.28%; (b) The measured accuracy loss of this prototype when reducing supply voltage with different R.

PE. And the corresponding convolution is performed. Here, for each of the 2x2 activations on the 64 channels, the PE takes one cycle to process. Since the following image data overlaps with the previously processed data at two locations, it only needs to wait for two clock cycles to send the following data, as shown in Fig. 7(b). All data reading logics are implemented through the control module, which only needs parameters to configure the input image size to calculate each step automatically.

III. Measurement Results

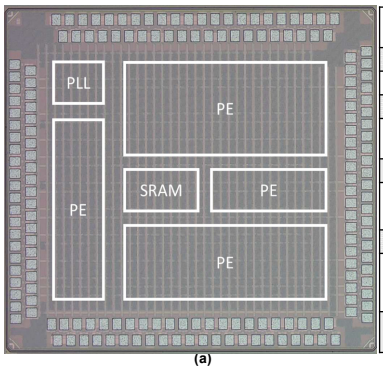
The SC-based NN processor prototype is fabricated in a 28nm CMOS process. It can operate at low voltages thanks to fault tolerance. It is measured with the operating frequency varying from 25MHz to 200MHz and the supply voltage varying from 0.575V to 0.8V. The chip's measured current consumption and energy efficiency are shown in Fig. 8. Across all frequencies, the lower the supply voltage, the higher the energy efficiency. The peak energy efficiency is 198.9 TOPS/W at 200MHz and 650mV.

To measure the accuracy, the TNN mentioned in Fig. 6 is loaded, and the software classification accuracy of the MNIST dataset is 98.28%. Note that both binary complement number and TCB use a 2-bit ternary representation. We simulated a binary design and compared it with the proposed SC design on the loss of classification accuracy under the same BER. Low supply voltages cause the increased BER. As shown in Fig. 9(a), when the accuracy of binary design is dropped to 85.34%, the accuracy of SC design

TABLE I
COMPARISON WITH SOTA BINARY-BASED NN PROCESSORS

	ISSCC'18[12]	ISSCC'19[13]	ISSCC'20[14]	JSSC'21[15]	ISSCC'21[16]	This Work
Process	65nm	8nm	7nm	28nm	28nm	28nm
Architecture	Binary	Binary	Binary	Binary	Binary	SC
Bit Precision	1-16	1-12	8	2-8	8	2
Die Area(mm²)	16	5.5	3.04	5.64	#1.9	4.5
On-chip SRAM(KB)	256	1568	2176	416	206	98
Voltage(V)	0.63-1.1	0.5-0.8	0.575-0.825	0.75-1.1	0.6-0.9	0.575-0.8
Frequency (MHz)	200	67-933	290-880	120-268	100-470	25-200
Power(mW)	3.2-297	39-1553	174-1053	6.75-36	19.4-131.6	4.6-59.8
*Energy Efficiency (TOPS/W)	11.6(4b) 50.69(1b)	11.5(8b)	13.32 (8b)	32.9(8b) 172(2b)	12.1(8b)	**198.9(2b)**
*Area Efficiency (GOPS/mm²)	86(4b) 460(1b)	347(8b) *1261(8b)	1186(8b)	389(2b) 24.3(8b)	#745.1(8b)	**2630(2b)**

* 1 OP = 1 addition or 1 multiplication
* 75% weight zeros
\# Area excluding SRAM array



(a)

Technology (nm)	28
Size (mm²)	2.00x2.25
SRAM (KB)	98
Voltage (V)	0.575-0.8
Frequency (MHz)	25-200
Speed (GOPS)	10084@200MHz
Power (mW)	59.8@200MHz, 0.8V
	4.6@25MHz, 0.575V
Data Width (bit)	2
Energy Efficiency (TOPS/W)	198.9 @200MHz, 0.65V (0.907uJ/Prediction)
Area efficiency (GOPS/mm²)	2630 @200MHz, 0.65V

(b)

Fig. 10. (a) The micrograph; (b) The technical specifications.

is 94.96%. On average, the proposed design reduces accuracy loss by 70% compared to its binary counterpart. It firmly demonstrates the excellent fault tolerance of the proposed fully SC-based NN processor.

The proposed PEs have no random fluctuations as in conventional SC designs. Thus, the measured classification accuracy is 98.28% at the rated voltage of 0.8V, which is consistent with the software classification accuracy. Across four operating frequencies, we measured the accuracy loss of this prototype when reducing supply voltage with different R, as shown in Fig. 9(b). The proposed processor can work properly at low voltages. The accuracy loss and the minimum support voltage are lower with a higher R. The results show that the SC-based NN processor is fault-tolerant for bit errors caused by decreased voltages. This test also illustrates that the fault tolerance increases as R increases. In this test, R is manually altered to reveal its relationship with the fault tolerance capability. In practice, R does not change during inference. This prototype demonstrates robust near-threshold computing with less than 1% accuracy loss at 0.575V and 25MHz, less than 100mV above the transistor V_{th} in this process. This makes near-threshold operation feasible.

Fig. 10(a) shows the micrograph of the prototype fabricated in the 28nm CMOS process. It occupies an active area of 4.5mm². Since this prototype is the first silicon-proven SC-based NN processor, we compare it to the state-of-the-art (SOTA) binary-based NN processors [14]-[18], as shown in Table I. The fabricated SC-based NN processor achieves 198.9 TOPS/W energy efficiency at 200MHz and 0.65V, an average of 10.75x (1.16x-to-17.30x) improvement over SOTA processors. After normalization of the process and bit precision, the improvement is 6.65x (1.02x-to-14.93x). The measured area efficiency considering on-chip SRAM is 2630 GOPS/mm², with an average of 4.20x (2.09x-to-6.76x) improvement over SOTA processors, or an average of 3.70x (1.60x-to-6.76x) improvement (normalized).

IV. CONCLUSION

This letter introduces a parallel, fully SC-based NN processor that achieves 198.9 TOPS/W energy efficiency and 2630 GOPS/mm² area efficiency. The fabricated 28nm processor is the first SC-based silicon prototype. All on-chip operations in this design adopt fault-tolerant TCBs, maximizing the fault tolerance of SC and demonstrating robust near-threshold computing. It realizes SOTA performance and shows the great potential for low-cost Internet of Things (IoT) NN processors.

REFERENCES

[1] H. Esmaeilzadeh et al., Dark silicon and the end of multicore scaling," *38th Annual International Symposium on Computer Architecture (ISCA)*, San Jose, CA, USA, 2011, pp. 365-376.
[2] P. N. Whatmough et al., "A 28nm SoC with a 1.2GHz 568nJ/prediction sparse deep-neural-network engine with >0.1 timing error rate tolerance for IoT applications," *2017 IEEE ISSCC*, pp. 242-243.
[3] P. N. Whatmough et al., "DNN Engine: A 28-nm Timing-Error Tolerant Sparse Deep Neural Network Processor for IoT Applications," in *IEEE JSSC*, vol. 53, no. 9, pp. 2722-2731, Sept. 2018.
[4] W. Qian et al., "An architecture for fault-tolerant computation with stochastic logic," *IEEE Transactions on Computers*, vol. 60, no. 1, pp. 93-105, Jan. 2011.
[5] Z. Xia et al., "Neural Synaptic Plasticity-Inspired Computing: A High Computing Efficient Deep Convolutional Neural Network Accelerator," *IEEE TCAS-I*, vol. 68, no. 2, pp. 728-740, Feb. 2021.
[6] J. Li et al., "Towards acceleration of deep convolutional neural networks using stochastic computing," *22nd ASP-DAC*, pp. 115-120, 2017.
[7] K. Kim et al., "Dynamic energy-accuracy trade-off using stochastic computing in deep neural networks," *53rd IEEE DAC*, 2016, pp. 1-6.
[8] Y. Liu et al., "An energy-efficient stochastic computational deep belief network," *DATE*, 2018, pp. 1175-1178.
[9] L. Deng et al., "GXNOR-Net: Training deep neural networks with ternary weights and activations without full-precision memory under a unified discretization framework," in *Neural Networks*, vol. 100, pp. 49–58, 2018.
[10] K. E. Batcher, "Sorting networks and their applications," *Spring Joint Computer Conference*, pp. 307-314, April 1968.
[11] S. Mohajer et al., "Routing magic: Performing computations using routing networks and voting logic on unary encoded data," *ACM/SIGDA International Symposium on FPGA*, pp. 77-86, 2018.
[12] Y. Lecun et al., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
[13] V. Sze et al., "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295-2329, Dec. 2017.
[14] J. Lee et al., "UNPU: A 50.6TOPS/W unified deep neural-network accelerator with 1b-to-16b fully-variable weight bit-precision," *IEEE ISSCC, 2018*, pp. 218-220.
[15] J. Song et al., "An 11.5TOPS/W 1024-MAC Butterfly Structure Dual-Core Sparsity-Aware Neural Processing Unit in 8nm Flagship Mobile SoC," *IEEE ISSCC*, 2019, pp. 130-131.
[16] C. Lin et al., "A 3.4-to-13.3TOPS/W 3.6TOPS Dual-Core Deep-Learning Processor for Versatile AI Applications in 7nm 5G Smartphone SoC," *IEEE ISSCC*, 2020, pp. 134-136.
[17] F. Tu et al., "Evolver: A Deep Learning Processor with On-Device Quantization–Voltage–Frequency Tuning," *IEEE JSSC*, vol. 56, no. 2, pp. 658-673, Feb 2021.
[18] H. Mo et al., "A 28nm 12.1TOPS/W Dual-Mode CNN Processor Using Effective-Weight-Based Convolution and Error-Compensation-Based Prediction," *IEEE ISSCC*, 2021, pp. 146-148.