Design of Median Filters in Stochastic Computing

Yongqiang Zhang, *Member, IEEE,* Defang Meng, Jie Han, *Senior Member, IEEE,* Shaowei Wang, Guangjun Xie, *Senior Member, IEEE*

***Abstract*—Median filtering is a nonlinear filtering technique to eliminate impulse noises from images by replacing each pixel with the median of its local neighboring pixel values. The design of median filters (MFs) in stochastic computing (SC) is mainly based on a sorting network to find the median among a set of pixels. This method has to sort values in multiple levels and will consume a large hardware cost. This work proposes a design method of MFs to obtain the median within a square window using majority (MAJ) gates with maximally correlated stochastic numbers (SNs). The MAJ gates can also be directly replaced by AND and OR gates. Experimental results show that the proposed MF saves area and power by up to 35.65% and 32.15% on average, respectively, compared to binary and best SC designs.**

Keywords—Median filter, stochastic computing, correlation, majority gate.

I. Introduction

Filtering is of importance in multiple areas, such as digital image and video processing, audio and speech processing, and wireless communication. Median filtering is a typical nonlinear filtering technique to eliminate impulse noises from images or video frames, such as salt-and-pepper noise [1, 2]. Besides, it can preserve sharp edges with higher efficiency than mean filtering and Gaussian filtering algorithms, which smooth images more than median filtering. The design of a median filter (MF) has been deeply studied, with the typical goal to sort sampling values in a sliding window and find the median among them. The central pixel value in the window is then replaced using the median [3]. The most famous MF is the order-statistic filter or named rank filter, through a full sort. Each sorting uses a compare and swap (CAS) module to compare the values of two pixels and swap them according to the design requirement. The implementation of a CAS module in a conventional binary manner, however, consumes a large hardware cost because of the complexity of a large number of digital comparators.

Stochastic computing (SC), as a reemerging computing paradigm, has the potential to lower circuit design complexity because of its weightless data encoding pattern, instead of the weighted binary computing (BC). Li has designed a CAS module with two inputs (CAS2) based on a stochastic tanh function [2]. The stochastic number (SN) generation for each input pixel value in this CAS2 module consumes too much power. To tackle this issue, the correlation between multiple SNs has been employed to reduce the number of consumed SN generation modules and optimize circuit complexity significantly [4]. Accordingly, the CAS2 module is optimized into a 2-input AND gate and a 2-input OR gate only [4]. These two gates directly generate the min and max values for inputs.

To optimize the structure of an MF and cater to emerging nanoelectronics, this work proposes a design method through majority (MAJ) gates with maximally correlated SNs. A 3-input MAJ gate with maximally correlated inputs can find the median among three inputs directly. This function is derived from the definition of SC correlation [5]. With it, a CAS module with three inputs (CAS3) is designed, which consists of a 3-input AND, a 3-input OR, and a 3-input MAJ, for sorting three values conveniently. Correspondingly, a fast sort algorithm and the structure of a 3×3 MF are introduced to compute the median and applied to several polluted images. This design is then evaluated and compared with previous designs with respect to computing accuracy and hardware costs.

This paper proceeds as follows. Section II briefly reviews the basics of SC. Section III details related works about MFs. Section IV illustrates the proposed design method. Section V shows experimental results in computing accuracy and hardware costs. Section VI concludes this work.

II. Basics

A. Stochastic Computing

A typical SC system has three parts, including a binary-to-stochastic (B2S) converter, also called stochastic number generator (SNG), a SC core for arithmetic computing, and a stochastic-to-binary converter (S2B) or derandomizer (DER), as shown in Fig. 1(a).

SC computes on SNs, each of which consists of unweighted logic 1s and 0s [6]. That is, the value that an SN represents is proportional to the number of 1s, given its length, instead of their positions. A binary number (BN) X within [0,1] can be represented by the probability p of each bit in an SN x being 1 in the unipolar format, i.e., $X=p(x_i=1)$. In the bipolar format, the value of a BN Y represented by an SN y is $2(y_i=1)-1$. Generally, an SN is generated through an SNG, which is composed of a random number source (RNS) and a

This work was supported in part by the National Natural Science Foundation of China under Grant 62404067, in part by the Fundamental Research Funds for the Central Universities under Grant JZ2025HGTB0231, and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Grant RES0048688.

(Corresponding author: Guangjun Xie)

Y. Zhang, D. Meng, and G. Xie are with the School of Microelectronics, Hefei University of Technology, Hefei 230009, China (e-mail: ahzhangyq@hfut.edu.cn; 2674502625@qq.com; gjxie8005@hfut.edu.cn)

J. Han is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: jhan8@ualberta.ca)

S. Wang is the School of Integrated Circuits, Anhui University, Hefei 230601, China (e-mail: swwang@ahu.edu.cn)

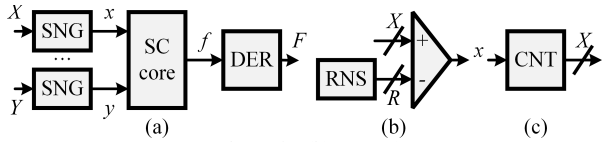


Fig. 1. An SC system. (a) Schematic. (b) SNG. (c) DER.

TABLE I. FUNCTIONS OF LOGIC GATES WITH DIFFERENT SCC VALUES

Gate	SCC		
	0	**+1**	**-1**
AND	XY	$\min(X,Y)$	$\max(X+Y-1,0)$
OR	$X+Y-XY$	$\max(X,Y)$	$\min(X+Y,1)$
XOR	$X+Y-2XY$	$\lvert X-Y\rvert$	$X+Y$, if $X+Y\le 1$; $2-(X+Y)$, if $X+Y>1$

comparator (CMP), as shown in Fig. 1(b). If a BN X is larger, or smaller, than a BN R from the RNS, the SNG produces a 1, or a 0, at each clock cycle. An RNS can be a linear feedback shift register (LFSR) [7], a Sobol or Halton sequence generator [8, 9], or even a binary counter (CNT) [10]. To convert a bitstream x back to a BN X, a DER is simply designed using a binary CNT to count the number of 1s in the bitstream serially in time [7], as shown in Fig. 1(c).

B. Stochastic Computing Correlation

As is known, an AND gate realizes the multiplication of SNs x and y, which is established on the premise of independence between the two SNs. If two SNs are correlated, the function of a gate may be weird, but useful [5]. As described, a well utilization of stochastic computing correlation (SCC) between two SNs can optimize the design complexity and improve computing accuracy of some operations, such as, absolute valued subtraction. The SCC between two SNs is defined as

$$\mathrm{SCC}(x,y)=\begin{cases}\dfrac{\delta(x,y)}{\min(p_x,p_y)-p_xp_y}, & \delta(x,y)>0\\ 0, & \delta(x,y)=0\\ \dfrac{\delta(x,y)}{p_xp_y-\max(p_x+p_y-1,0)}, & \delta(x,y)<0\end{cases}, \quad (1)$$

where $\delta(x,y)=p_{xy}-p_xp_y$, p_x and p_y are the probabilities denoted by SNs x and y, and p_{xy} is the probability of the ANDed results of x and y. SCC=0 indicates x and y are ideally independent, while SCC=+1 or -1 means a maximally positive or negative correlation. TABLE I lists the functions of AND, OR, and XOR gates using input bitstreams with different SCCs in the unipolar format. AND and OR gates with uncorrelated bitstreams carry out multiplication and imperfect addition, respectively. An XOR with two positively correlated bitstreams can easily realize absolute valued subtraction [5].

III. RELATED WORKS

Noise reduction through an MF is mainly to remove impulse noise and preserve edges. The key to designing an MF is to find the median among the neighboring pixel values of a central pixel. Take a 3×3 MF as an example. A typical structure of MFs is based on a sorting network, as shown in Fig. 2, where I_1~I_9 are 9 input pixel values to be sorted, and

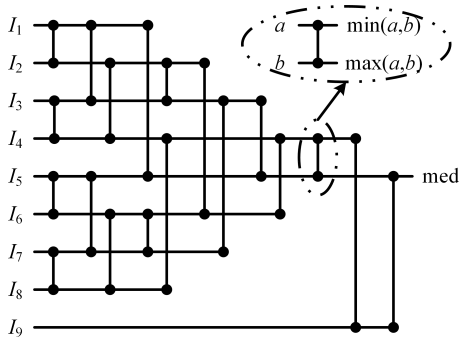


Fig. 2. A 3×3 MF based on CAS2 modules.

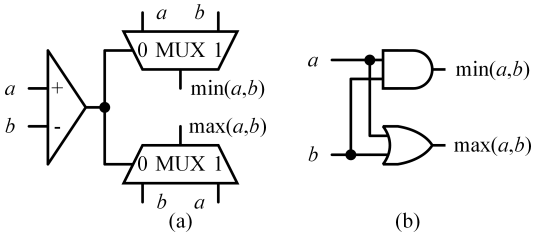


Fig. 3. CAS2 module. (a) Binary implementation. (b) Stochastic implementation based on SN correlation.

med is the median value among them. The inset surrounded by a dashed ellipse is a CAS2 module to find the min and max of a and b. Thus, a 3×3 MF needs 19 CAS2 modules.

Each CAS2 can be implemented in different ways, such as a weighted binary design shown in Fig. 3(a), where a and b are inputs to be compared and sorted, $\min(a,b)$ and $\max(a,b)$ are generated outputs. Note that both the CMP and multiplexers (MUXs) are M-bit, where M is the bitwidth of inputs. Intuitively, this design will require higher hardware area and power consumption for larger bitwidth inputs.

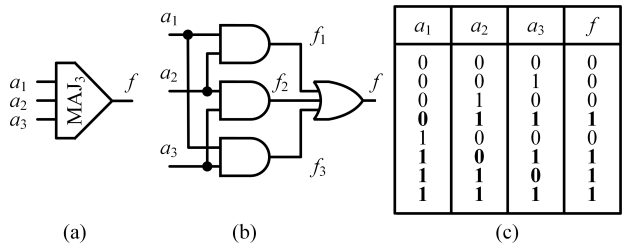
A stochastic CAS2 consists of a stochastic CMP and two 1-bit MUXs to generate the minimum and maximum values. The inputs in the first level of an MF are transformed into SNs using different SNGs. The detailed mathematical proof can be found in [11].

As for the two CMPs above, although the computing core of the latter design is simpler than the first one, it needs two SNGs to generate independent SNs to ensure correct logic functions. As described in [7], SNGs could occupy more than 80% area in some applications. To reduce the number of SNGs, Fig. 3(b) illustrates a stochastic CAS2 using positive SN correlation, which employs only an AND gate and an OR gate [4]. As listed in TABLE I, these two gates can compute $\min(a,b)$ and $\max(a,b)$. Note that this computing core is not only such simpler, but also just needs one RNS shared between two SNGs for generating two maximally correlated SNs.

IV. THE PROPOSED DESIGN METHOD

A. Three-input Majority Gate

Some emerging nanoelectronics, such as spintronic threshold device (spin-TD) [12], nanomagnetic logic (NML) [13], and quantum-dot cellular automata (QCA) [14], rely on an MAJ gate as their design primitive, instead of AND and OR gates. An MAJ has n inputs, where n is a positive odd integer,



a_1	a_2	a_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	**1**	**1**	**1**
1	0	0	0
1	**0**	**1**	**1**
1	**1**	**0**	**1**
1	**1**	**1**	**1**

Fig. 4. A 3-input MAJ. (a) Symbol. (b) Implementation using AND and OR gates. (c) Truth table.

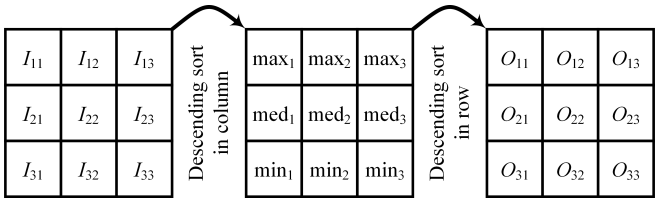


Fig. 5. The procedure to find the median in a 3×3 window.

such as 3, 5, and 7 [12]. The output is determined by the majority of its Boolean logic inputs. For example, a logic 1 is generated if and only if more than $\lfloor n/2 \rfloor$ of the logic inputs are 1. The symbol of a 3-input MAJ is shown in Fig. 4(a), where a_1, a_2, and a_3 are logic inputs, and f is a logic output. The function can also be realized using 3 2-input AND gates and a 3-input OR gate in CMOS technology, as shown in Fig. 4(b). Its truth table is shown in Fig. 4(c). Thus, the Boolean logic function of a 3-input MAJ is written as

$$F = \text{MAJ}(a_1, a_2, a_3) = a_1a_2 + a_1a_3 + a_2a_3 . \quad (2)$$

Consider transforming it to an arithmetic function in the unipolar format here. The function is written as

$$p_f = p_{a_1a_2} + p_{a_1a_3} + p_{a_2a_3} - 2p_{a_1a_2a_3} , \quad (3)$$

where a_1, a_2, a_3, and f are interpreted as SNs. If the three inputs are independent of each other, the output is

$$F = A_1A_2 + A_2A_3 + A_1A_3 - 2A_1A_2A_3 , \quad (4)$$

where $A_1 = p_{a_1}$, $A_2 = p_{a_2}$, $A_3 = p_{a_3}$, and $F = p_f$ are binary values within [0,1]. A_3 can be set to different values to realize various arithmetic functions. For example, if A_3=0, F=A_1A_2; if A_3=1/2, F=$(A_1+A_2)/2$. Stochastic multiplication and scaled subtraction are implemented by configuring the value of A_3. If the three inputs are maximally correlated to each other, that is, SCC(a_1,a_2)=1, SCC(a_2,a_3)=1, and SCC(a_1,a_3)=1, the output has to be considered according to the numerical values of inputs. Suppose A_3<A_2<A_1. According to (1), one has $p_{a_1a_2} = p_{a_2}$, $p_{a_1a_3} = p_{a_3}$, and $p_{a_2a_3} = p_{a_3}$. As for $p_{a_1a_2a_3}$, it can be computed as

$$p_{a_1a_2a_3} = p_{a_1a_2/a_3} p_{a_3} , \quad (5)$$

where $p_{a_1a_2/a_3}$ is a conditional probability that the probability of $p_{a_1a_2}$ under the assumption of a_3=1. Since a_1, a_2, and a_3 are maximally correlated and A_3<A_2<A_1, $p_{a_1a_2/a_3} = 1$. Thus, $p_{a_1a_2a_3} = p_{a_3}$. The other cases can be derived in the same way.

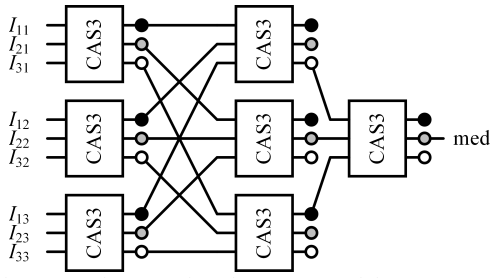


Fig. 6. The proposed 3×3 MF based on CAS3 modules.

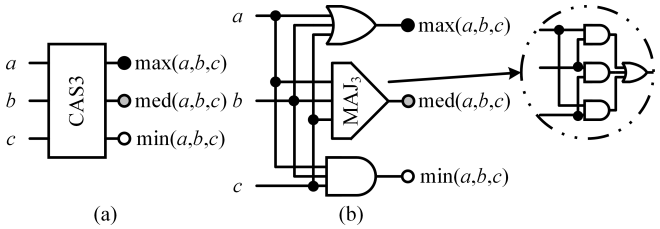


Fig. 7. CAS3 module. (a) Symbol. (b) Implementation.

B. The Proposed Median Filter

The size of a sliding window to filter an image can be 3×3, 5×5, or even 7×7. The larger the sliding window is, the more the hardware costs of an MF are. However, the processed image quality may not be linear to the window size. Here, we consider a sliding window size of 3×3 as in previous works [1, 2, 8, 10, 15].

Fig. 5 illustrates the procedure to find the median value among 9 pixel values I_{ij} (i = 1, 2, and 3, j = 1, 2, and 3) in a 3×3 sliding window. The first step is to make a descending sort for each of the three columns to generate B_{1j}, B_{2j}, and B_{3j}. Thus, $B_{1j} \geq B_{2j} \geq B_{3j}$ in each column. Then, a descending sort is made for B_{ij} for each row to generate O_{ij}. Thus, $O_{i1} \geq O_{i2} \geq O_{i3}$, in each row. The numerical values of O_{ij} can be compared as follows.

O_{11} is the largest value among the 9 pixel values because it is the max among the three max values, B_{11}, B_{12}, and B_{13}. Similarly, O_{33} is the least value among them. O_{12} is the median value of O_{11}, O_{12}, and O_{13}, and represented as O_{12} = median(O_{11}, O_{12}, O_{13}) = $\max_i \geq \text{median}_i \geq \min_i$. It is also larger than the value of O_{13} = min(O_{11}, O_{12}, O_{13}) = $\max_j \geq \text{median}_j \geq \min_j$. Thus, O_{12} is larger than five values at least. Similarly, O_{32} is less than five values at least. As for O_{21}, O_{22}, and O_{23}, they are represented as $O_{21} = \text{median}_i \geq \min_i$, $O_{22} = \text{median}_j \geq \min_j$, $O_{23} = \text{median}_k \geq \min_k$, and $O_{21} \geq O_{22} \geq O_{23}$ (k = 1, 2, and 3). Thus, O_{21} is larger than five values, and O_{23} is less than five values at least, respectively. With this analysis, since O_{11}, O_{12}, O_{21}, O_{23}, O_{32}, and O_{33} can not be the median value, it is a number within O_{13}, O_{22}, and O_{31}.

According to Fig. 5, we propose a design method of an MF, as shown in Fig. 6, which consists of 7 CAS3 modules to find max, median, and min values in parallel. It can finish operation in three levels, instead of 8 levels in Fig. 2. The symbol of the CAS3 is shown in Fig. 7(a), which is composed of 3-input AND and OR gates for generating the max and min values and a 3-input MAJ for the median value in Fig. 7(b). This design operates on SNs with maximally positively correlated SNs. So, an SNG is shared with inputs a, b, and c. In addition, the

TABLE II. HARDWARE COSTS OF 3×3 MFS

Designs	Area			CPD	Power	ADP	PDP
	Total	SNG	Core				
Binary	15.53	/	15.53	8.44	7.11	13.10	59.97
SMF_{SCC}	5.22	3.13	0.82	1.40	2.60	0.73	3.64
SMF_{MAJ}	5.03	3.13	0.63	1.26	2.59	0.63	3.26

Area: $10^2\mu m^2$; CPD: ns; Power: 10^2mW; ADP: $10^3\mu m^2$·ns; PDP: 10^{-2}mW·ns.

TABLE III. PSNR (DB) AND MSSIM VALUES OF 3×3 MFS WITH VARIOUS NOISE DENSITIES

Designs	Binary	SMF_{SCC}	SMF_{MAJ}
PSNR/MSSIM	30.43/0.8689	30.43/0.8689	/0.8689

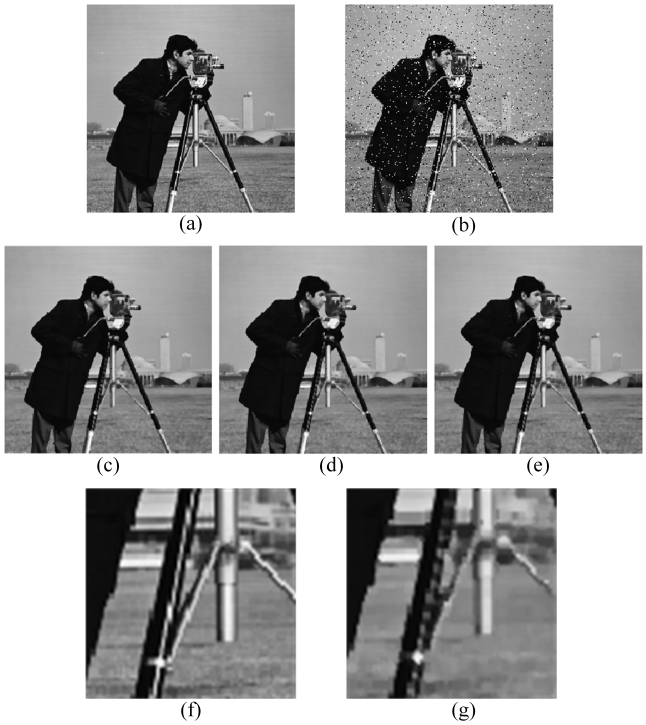


Fig. 8 Experimental images. (a) Original image. (b) Polluted image. (c) Binary. (d) SMF_{SCC}. (e) SMF_{MAJ}. (f) Local enlarged view of the original image. (g) Local enlarged view of the SMF_{MAJ} filtered image.

proposed MF also needs only an SNG. In SMF_{MAJ}, all SNs are generated synchronously from a single RNS. No additional delay, reordering, or bit scrambling is applied in multi-level operations, and all SNs are processed in bit-wise alignment. Therefore, the SCC can be well preserved throughout the cascaded logic stages. Note that the 3-input MAJ can also be replaced with AND and OR gates, as in Fig. 4(b), if the 3-input MAJ of a considered nanotechnology is still up in the air in physical implementation at present.

V. EXPERIMENTAL RESULTS

The proposed MF in Fig. 6 is compared with typical MFs based on a sorting network in both hardware overhead and image processing. Among them, there are two typical realizations for the CAS2 modules as shown in Fig. 3. All circuits are described in Verilog hardware description language (HDL) and functionally simulated using Modelsim. The hardware costs are tested with Synopsys Design Compiler using a 65nm process technology at the typical corner. The operating frequency is set to 100 MHz. For real image processing, four images, including cameraman, moon, clock, and airplane downloaded from the website [16], are polluted and then filtered using these three MFs in MATLAB. The density of the added noise is set to 0.05. The employed RNS is an 8-bit linear feedback shift register (LFSR) with the same seed, and the bitwidth of the binary MF to be compared is 8.

A. Hardware Costs

TABLE II summarizes the hardware results of the three MFs, including area, critical path delay (CPD), power consumption, along with area-delay product (ADP) and power-delay product (PDP). The area overhead of the two SC designs is divided into SNG and computing core areas. As shown in TABLE II, compared with other designs, the proposed MF has achieved optimization in all indicators, except for the same SNG area as the correlation based MF. For example, the core area has been reduced by at least 23.25%, and the CPD has been decreased by at least 10.00%, thereby optimizing both ADP and PDP.

B. Image Processing

The mean values of peak signal-to-noise ratio (PSNR) and mean structural similarity index measure (MSSIM) for four filtered images under various noise densities are shown in TABLE III. The results indicate that the proposed MF can maintain the same results as the existing designs. Since a 3×3 MF is just to select a value from 9 inputs to replace the central pixel, they generate the same results. The original and processed 256×256 grayscale cameraman images using three MFs, with an added noise density of 0.05, are shown in Fig. 8.

VI. CONCLUSION

In this work, a sorting method for finding the median value among a 3×3 window is described. To realize this algorithm, a CAS3 module is proposed based on a 3-input AND, a 3-input OR, and a 3-input MAJ with maximally positively correlated SNs. Physical synthesis results demonstrate that the proposed MF based on the CAS3 modules has a significant improvement in hardware costs compared to previous counterparts, while preserving the same processing results. For example, the occupied area is reduced by up to 35.65% on average, compared with two previous designs. The discussion for 5×5 and 7×7 MFs will be the main focus of our future work, which will be further elaborated in our subsequent extended research.

REFERENCES

[1] S. Asadi, A. Jalilvand, M. Najafi, and M. Bayoumi, "ECO: Enhanced in-stream correlation manipulation for low-discrepancy stochastic computing," *IEEE Trans. Very Large Scale Integr. VLSI Syst.,* vol. 33, no. 11, pp. 3085-3096, Nov. 2025.

[2] P. Li, D. Lilja, W. Qian, K. Bazargan, and M. Riedel, "Computation on stochastic bit streams digital image processing case studies," *IEEE Trans. Very Large Scale Integr. VLSI Syst.,* vol. 22, no. 3, pp. 449-462, Mar. 2014.

[3] R. Gonzalez, and R. Woods, *Digital image processing*: Pearson, 330 Hudson Street, New York, NY 10013, 2018.

[4] R. Budhwani, R. Ragavan, and O. Sentieys, "Taking advantage of correlation in stochastic computing," in 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 2017, pp. 1-4.

[5] A. Alaghi, and J. Hayes, "Exploiting correlation in stochastic circuit

design," in 2013 IEEE 31st International Conference on Computer Design (ICCD), Asheville, NC, USA, 2013, pp. 39-46.
[6] Y. Zhang, X. Chen, J. Han, and G. Xie, "Stochastic mean circuits based on inner-product units using correlated bitstreams," *IEEE Trans. Circuits Syst., II, Exp. Briefs,* vol. 71, no. 2, pp. 867-871, Feb. 2024.
[7] W. Qian, X. Li, M. Riedel, K. Bazargan, and D. Lilja, "An architecture for fault-tolerant computation with stochastic logic," *IEEE Trans. Comput.,* vol. 60, no. 1, pp. 93-105, Jan. 2011.
[8] S. Liu, and J. Han, "Toward energy-efficient stochastic circuits using parallel Sobol sequences," *IEEE Trans. Very Large Scale Integr. VLSI Syst.,* vol. 26, no. 7, pp. 1326-1339, Jul. 2018.
[9] Z. Lin, G. Xie, W. Xu, J. Han, and Y. Zhang, "Accelerating stochastic computing using deterministic Halton sequences," *IEEE Trans. Circuits Syst., II, Exp. Briefs,* vol. 68, no. 10, pp. 3351-3355, Oct. 2021.
[10] M. Najafi, D. Lilja, M. Riedel, and K. Bazargan, "Low-cost sorting network circuits using unary processing," *IEEE Trans. Very Large Scale Integr. VLSI Syst.,* vol. 26, no. 8, pp. 1471-1480, Aug. 2018.
[11] P. Li, and D. Lilja, "Using stochastic computing to implement digital image processing algorithms," in the 2011 IEEE 29th International Conference on Computer Design (ICCD), Amherst, MA, USA, 2011, pp. 154-161.
[12] H. Jiang, S. Angizi, D. Fan, J. Han, and L. Liu, "Non-volatile approximate arithmetic circuits using scalable hybrid spin-CMOS majority gates," *IEEE Trans. Circuits Syst. I-Regul. Pap.,* vol. 68, no. 3, pp. 1217-1230, Mar. 2021.
[13] T. Zhang, H. Jiang, H. Mo, W. Liu, F. Lombardi, L. Liu, and J. Han, "Design of majority logic-based approximate Booth multipliers for error-tolerant applications," *IEEE Trans. Nanotechnol.,* vol. 21, pp. 81-89, Feb. 2022.
[14] Y. Zhang, C. Zhu, X. Cheng, and G. Xie, "Design and implementation of SRAM for LUT and CLB using clocking mechanism in quantum-dot cellular automata," *IEEE Trans. Circuits Syst., II, Exp. Briefs,* vol. 69, no. 9, pp. 3909-3913, May. 2022.
[15] N. Temenos, and P. Sotiriadis, "Stochastic computing max & min architectures using Markov chains: Design, analysis, and implementation," *IEEE Trans. Very Large Scale Integr. VLSI Syst.,* vol. 29, no. 11, pp. 1813-1823, Nov. 2021.
[16] "The USC-SIPI image database." 1981. [Online]. Available: https://sipi.usc.edu/database/.