

Stochastic Mean Circuits Based on Inner-Product Units Using Correlated Bitstreams

Yongqiang Zhang, *Member, IEEE*, Xiaoyue Chen, Jie Han, *Senior Member, IEEE*, and Guangjun Xie, *Senior Member, IEEE*

Abstract—Stochastic mean circuits (SMCs) are at the core of mean filtering and neural networks, but they have not been fully investigated in stochastic computing (SC). In this paper, SMCs in the unipolar and bipolar formats are respectively proposed to average bitstreams. For this purpose, the unipolar and bipolar stochastic inner-product circuits (SIPCs) are first designed by using positively correlated bitstreams for multiplication and accumulation. These bitstreams are generated by sharing a random number source (RNS) among different stochastic number generators (SNGs) to lower hardware costs. Applications of the proposed designs in Gaussian filtering, edge detection, and mean filtering are implemented. Experimental results show that the proposed SIPCs and SMCs outperform previous designs in both hardware efficiency and computing accuracy.

Index Terms—Stochastic computing, correlation, inner-product circuit, mean circuit.

I. INTRODUCTION

STOCHASTIC computing (SC) encodes binary numbers into randomly distributed bitstreams when a slight computing inaccuracy is acceptable in applications such as image processing. The bitstreams make it feasible for simple logic gates to implement complex functions. For example, an AND gate or a multiplexer (MUX) performs the multiplication or addition of two independent bitstreams [1].

Another basic component in SC is the stochastic mean circuit (SMC), which is at the core of filtering and neural networks [2]. One way to average stochastic bitstreams is to use MUXs by controlling the select bitstreams [3]. Although the input bitstreams to be averaged can be generated by sharing one random number source (RNS), the select bitstreams have to be independently produced to lower the stochastic computing correlation (SCC), thus causing a large hardware cost. A unipolar SMC has been proposed to average positively correlated bitstreams [4]. The design uses an RNS to generate correlated bitstreams to avoid the stringent requirement of decorrelation. However, minimum detectors are used, making the SMC inevitably expensive in hardware implementation. Other designs include 2-, 4-, and 9-input unipolar SMCs to average positively correlated bitstreams in [5]. Although these designs use two RNSs for respectively generating coefficient

and input bitstreams, preliminary results validate their advantage over previous designs. This work is a thorough extension of [5]; it provides two general design methods of stochastic inner-product circuits (SIPCs) and SMCs in the unipolar and bipolar formats with additional theoretical analysis and applications. The SIPC and SMC are applicable to any number of inputs and consume only one RNS. The main contributions of this work are as follows. (1) Two SIPCs are proposed for positively correlated bitstreams in the unipolar and bipolar formats, respectively. (2) Two SMCs based on the SIPCs are proposed to average positively correlated bitstreams in the unipolar and bipolar formats, respectively. (3) Theoretical proofs are provided for the proposed designs. (4) Applications in Gaussian filtering, Sobel edge detection, and mean filtering are implemented.

This paper proceeds as follows. Section II provides the basics of SC. Section III briefly reviews related works. Section IV illustrates the proposed SIPCs and SMCs. Section V provides the experimental results and applications in image processing. Section VI concludes this work.

II. BASICS

Stochastic numbers or the so-called bitstreams are generated by a stochastic number generator (SNG), which is generally composed of an RNS and a comparator. A number R generated by the RNS is compared to a k -bit binary number B at every clock cycle to produce a bitstream b with a length of 2^k bits, as shown in the 2nd column in TABLE I. If B is larger, or smaller, than R , the SNG produces a 1, or a 0, at each clock cycle. The RNS can be a linear feedback shift register (LFSR) [6], a Sobol sequence generator (SSG) [7], or other pseudorandom number generators. In the unipolar format, the probability p of each bit being 1 in the generated bitstreams equals the corresponding binary number B within $[0, 1]$, while the probability p is $(A+1)/2$ in the bipolar format, where A is within $[-1, 1]$. We use lowercases to indicate bitstreams and uppercases to denote corresponding binary numbers, unless otherwise specified.

The scaled adder and multiplier are two basic components in SC, as respectively listed in the 3rd, 4th, and 5th columns in

This work was supported in part by the Fundamental Research Funds for the Central Universities of China under Grant JZ2020HGQA0162; and in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under Grant RES0048688. (*Corresponding author: Guangjun Xie*)

Yongqiang Zhang, Xiaoyue Chen, and Guangjun Xie are with the School of Microelectronics, Hefei University of Technology, Hefei 230009, China (e-mail: ahzhangyq@hfut.edu.cn; 1617090911@qq.com; gjxie8005@hfut.edu.cn)

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

TABLE I
STOCHASTIC COMPONENTS

Operation	(a) SNG	(b) Unipolar/Bipolar addition	(c) Unipolar multiplication	(d) Bipolar multiplication	(e) Addition	(f) Absolute subtraction
Component						
Function	$B \rightarrow b$	$F = SX + (1-S)Y$	$F = XY$	$F = XY$	$F = X + Y$	$F = X - Y $
Correlation	NA	Uncorrelated	Uncorrelated	Uncorrelated	Negatively correlated	Positively correlated

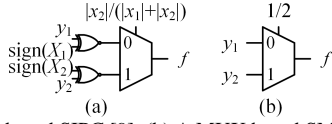


Fig. 1. (a) A MUX based SIPC [9]. (b) A MUX based SMC.

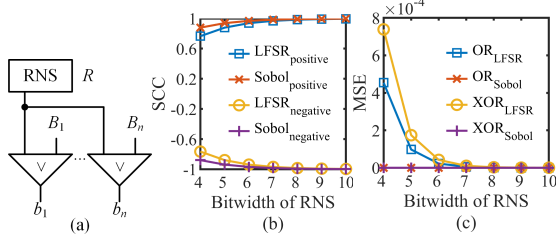


Fig. 2. (a) The generation of positively correlated bitstreams adapted from [11]. (b) The SCC values of sharing an LFSR and SSG. (c) The MSEs of OR and XOR gates performing addition and absolute subtraction.

TABLE I. The bitstream s of the adder has to be uncorrelated with x and y to ensure a correct function $SX+(1-S)Y$, while x and y can be correlated with each other. This adder is feasible in both unipolar and bipolar formats. An AND or XNOR gate performs the multiplication in the unipolar or bipolar format, respectively, if the bitstreams x and y are uncorrelated. Yet, an OR gate with negatively correlated bitstreams can perform the saturated addition $X+Y$ if $X+Y \leq 1$ or 1 if $X+Y > 1$; an XOR gate with positively correlated bitstreams realizes the absolute subtraction $|X-Y|$ [8], as in the last two columns in TABLE I.

III. RELATED WORK

A. Stochastic Inner-Product Circuits (SIPCs)

A bipolar SIPC is based on a MUX [9], as shown in Fig. 1(a). The bitstreams y_1 , y_2 , and $|x_2|/(|x_1|+|x_2|)$ are generated by three SNGs, respectively corresponding to binary numbers Y_1 , Y_2 , and $|X_2|/(|X_1|+|X_2|)$. It performs the scaled inner-product function, as

$$F = \frac{X_1 Y_1 + X_2 Y_2}{|X_1| + |X_2|} = \frac{\text{sign}(X_1) |X_1| Y_1 + \text{sign}(X_2) |X_2| Y_2}{|X_1| + |X_2|}, \quad (1)$$

where $\text{sign}(X_i)$ ($i=1$ or 2) is 1 if $X_i \geq 0$, or $\text{sign}(X_i)$ is -1. To lower the hardware costs of generating multiple bitstreams, circular shifting of the bits of an RNS has been considered to share the RNS within multiple SNGs [10]. This method works well for SIPCs with fewer inputs.

B. Stochastic Mean Circuits (SMCs)

By setting x_1 and x_2 to be 1 for the SIPC, a MUX based SMC to average y_1 and y_2 can be obtained, as shown in Fig. 1(b). Multi-input SMCs have been designed and evaluated [3], for which multiple uncorrelated select bitstreams have been generated by sharing an RNS to save hardware. Positively correlated bitstreams can be averaged by simple logic gates; meanwhile, a high computing accuracy can also be preserved, as demonstrated in [4]. Minimum detectors to distinguish the smallest value cause large hardware costs, which makes the method inferior.

IV. THE PROPOSED DESIGNS

A. Positively Correlated Bitstreams

Logic gates using correlated bitstreams can perform specific functions, as described above. The SCC value of two bitstreams x and y is defined as [8]

$$\text{SCC}(x, y) = \begin{cases} \frac{\delta(x, y)}{\min(p_x, p_y) - p_x p_y}, & \delta(x, y) > 0 \\ 0, & \delta(x, y) = 0, \\ \frac{\delta(x, y)}{p_x p_y - \max(p_x + p_y - 1, 0)}, & \delta(x, y) < 0 \end{cases} \quad (2)$$

where $\delta(x, y) = p_{xy} - p_x p_y$ (p_{xy} is the probability of the ANDed results of x and y). $\text{SCC}=0$ means two bitstreams are ideally independent, while a value of +1 or -1 respectively indicates a maximally positive or negative correlation. The generation of n positively correlated bitstreams can be realized by sharing an RNS among n SNGs, as shown in Fig. 2(a) [11]. Assume n k -bit binary number $B_1, \dots, B_i, \dots, B_j, \dots, B_n$, and $B_i < B_j$. The generated bitstreams respectively are $b_1, \dots, b_i, \dots, b_j, \dots, b_n$, with a length of 2^k bits. In the t^{th} clock cycle ($1 \leq t \leq 2^k$), if $R < B_i$, the t^{th} bit of b_i is a 1, and the t^{th} bit of b_j is also a 1, and vice versa. Thus, the positions of 1 and 0 in b_i and b_j are overlapped to the extent possible. Consider a case, for example, $b_i=00001111$ and $b_j=11001111$. $\delta(b_i, b_j) = p_{b_i b_j} - p_{b_i} p_{b_j} = 4/8 - 4/8 \times 6/8 = 1/8$. Their $\text{SCC}(b_i, b_j)$ is 1, so b_i and b_j are positively correlated bitstreams.

Fig. 2(b) shows the average SCC values of two positively and negatively correlated bitstreams for different bitwidths of LFSRs and SSGs. The SNGs using SSGs consistently realize larger absolute SCC values than those using LFSRs. The absolute SCC values of two designs with 8-bit RNSs are 0.9844 and 0.9923, respectively, thus resulting in a high computing accuracy. This can be seen from the mean squared error (MSE) values in Fig. 2(c), where an OR or XOR gate respectively works with negatively or positively correlated bitstreams. Both gates reach very low MSE values. In fact, the MSE values for the OR and XOR gates using SSGs are 0, while those using LFSRs result in negligible accuracy loss because an LFSR usually does not have an all-zero state [12].

B. SIPCs

A SIPC performs the multiplication and addition as

$$F = \sum_{i=1}^n X_i Y_i = \sum_{i=1}^n \text{sign}(X_i) |X_i| Y_i, \quad (3)$$

where $X=(X_1, X_2, \dots, X_n)$ and $Y=(Y_1, Y_2, \dots, Y_n)$ are two binary vectors; $\text{sign}(X_i)$ is +1 or -1 for positive or negative numbers. The resulting F that exceeds 1 or -1 should be scaled down to $[0, 1]$ or $[-1, 1]$, respectively, corresponding to the unipolar or

TABLE II
RELATIONSHIPS OF THREE BITSTREAMS

Bitstream	Value			
$x_1 + \dots + x_i$	1			0
$x_1 + \dots + x_j$	1	0	0	0
$x_1 + \dots + x_k$	1	0	0	0

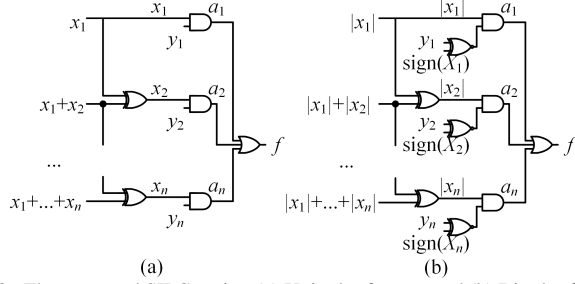


Fig. 3. The proposed SIPCs using (a) Unipolar format, and (b) Bipolar format. The maximum and minimum values of the unipolar and bipolar SIPCs have to be determined in advance, respectively, as (4) and (5)

$$[F_{\min}^u, F_{\max}^u] = \left[0, \sum_{i=1}^n X_i \right]. \quad (4)$$

$$[F_{\min}^b, F_{\max}^b] = \left[-\sum_{i=1}^n |X_i|, \sum_{i=1}^n |X_i| \right]. \quad (5)$$

Thus, the scaled results of the unipolar and bipolar SIPCs are respectively given in (6) and (7)

$$F^u = \sum_{i=1}^n \frac{X_i}{F_{\max}^u} Y_i. \quad (6)$$

$$F^b = \sum_{i=1}^n \frac{\text{sign}(X_i) |X_i|}{F_{\max}^b} Y_i. \quad (7)$$

Unipolar SIPCs: Fig. 3(a) shows the proposed unipolar SIPC. $x_1, x_1+x_2, \dots, \text{ and } x_1+\dots+x_n$ represent positively correlated bitstreams generated by sharing an RNS among n SNGs as in Fig. 2(a), respectively denoting scaled numbers $X_1/(X_1+\dots+X_n), (X_1+X_2)/(X_1+\dots+X_n), \dots, \text{ and } (X_1+\dots+X_n)/(X_1+\dots+X_n)$. The XORed results of the bitstreams are $x_1, x_2, \dots, \text{ and } x_n$ since the XORed results perform the absolute subtraction. $x_1, x_2, \dots, \text{ and } x_n$ are negatively correlated with each other and proved as follows.

The bitstreams x_i and x_j being negatively correlated mean $\text{SCC}(x_i, x_j) = -1$. Assume $k < j < i$ and $i-j+1=k+2$ for simplicity. According to (6) $p_{x_1} + \dots + p_{x_n} = 1, \max(p_{x_i} + p_{x_j}, -1, 0) = 0$. The binary numbers of bitstreams $x_1+\dots+x_k, x_1+\dots+x_j, \text{ and } x_1+\dots+x_i$ satisfy $(X_1+\dots+X_k)/(X_1+\dots+X_n) \leq (X_1+\dots+X_j)/(X_1+\dots+X_n) \leq (X_1+\dots+X_i)/(X_1+\dots+X_n)$. Thus, if $x_1+\dots+x_i$ is 1, $x_1+\dots+x_j$ can be 1 or 0, or $x_1+\dots+x_j$ and $x_1+\dots+x_k$ must be 0. If $x_1+\dots+x_j$ is 1, $x_1+\dots+x_k$ can be 1 or 0; otherwise, $x_1+\dots+x_k$ must be 0, as listed in TABLE II. If the values are 1, 1, and 1, the XORed results are $x_i=0$ and $x_j=0$; if the values are 1, 1, and 0, $x_i=0$ and $x_j=1$; if the values are 1, 0, and 0, $x_i=1$ and $x_j=0$; and if the values are 0, 0, and 0, $x_i=0$ and $x_j=0$. Thus, $p_{x_i, x_j} = p(x_i=1 \cap x_j=1) = 0$ and $\delta(x_i, x_j) = p_{x_i, x_j} - p_{x_i} p_{x_j} = -p_{x_i} p_{x_j}$, and then $\text{SCC}(x_i, x_j) = -1$.

The bitstreams y_1, \dots, y_n are generated by sharing an RNS with an inversed order [6]. The n AND gates perform the multiplication since the bitstreams x_i and y_i are uncorrelated. The ANDed results $a_1, \dots, \text{ and } a_n$ are also negatively correlated. It is proved in (8) using a_i and a_j .

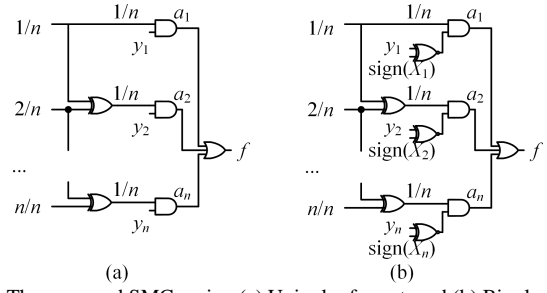


Fig. 4. The proposed SMCs using (a) Unipolar format, and (b) Bipolar format.

$$\begin{aligned} p_{a_i, a_j} &= p(a_i a_j = 1) = p(a_i = 1 \cap a_j = 1) \\ &= p(x_i y_i = 1 \cap x_j y_j = 1) = p(x_i y_i x_j y_j = 1) = 0 \end{aligned} \quad (8)$$

The last OR gate realizes the addition of n negatively correlated bitstreams $a_1, \dots, \text{ and } a_n$.

Bipolar SIPCs: Fig. 3(b) shows the proposed bipolar SIPC. If $\text{sign}(x_i)$ is +1, the connected pin is fixed to 1, while if $\text{sign}(x_i)$ is -1, the pin is 0. The XNOR gates realize the multiplication of y_i and $\text{sign}(x_i)$ in the bipolar format. Its working principle is similar to the one in the unipolar format.

C. SMCs

Fig. 4(a) shows the proposed unipolar SMC. It is realized by setting the inputs $X_1/(X_1+\dots+X_n), (X_1+X_2)/(X_1+\dots+X_n), \dots, \text{ and } (X_1+\dots+X_n)/(X_1+\dots+X_n)$ of the unipolar SIPC to be $1/n, 2/n, \dots, \text{ and } n/n$, respectively. Fig. 4(b) shows the proposed bipolar SMC with various binary inputs. It is necessary to make sure that their absolute values are $1/n, 2/n, \dots, \text{ and } n/n$. If all of them are set to positive values, the bipolar SMC is the same as the unipolar one, except for the different coding methods.

V. APPLICATIONS AND RESULTS

The proposed and compared stochastic designs in [3-5, 10] are modeled in MATLAB through m Monte Carlo trials for evaluating computing accuracy using the MSE defined as

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^m (f - f')^2, \quad (9)$$

where f and f' are respectively the exact and computed results using a stochastic circuit. Ten thousand Monte Carlo trials for each input combination are carried out to dampen the fluctuation of bitstreams. The proposed stochastic designs are implemented in Verilog and synthesized by the Synopsys Design Compiler with a TSMC 40 nm standard library at the typical corner [13]. The stochastic designs in [3-5, 10] are reprogrammed using the same design flow and implemented in the same technology node, as the proposed designs, to fairly judge their hardware performance. The command 'compile_ultra' is used to ungroup all components and automatically synthesize the circuits according to timing constraints. The clock period is set to be 10 ns. The averaged power is measured by the PrimeTime using a vector-based power model for ten thousand random input combinations. A vector is randomly generated using a uniform distribution in MATLAB to generate a *.vcd file in ModelSim for each design.

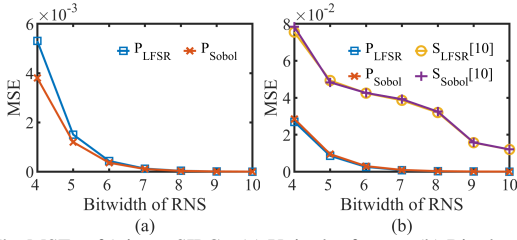


Fig. 5. The MSEs of 9-input SIPCs. (a) Unipolar format. (b) Bipolar format.

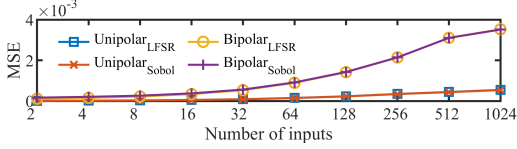


Fig. 6. The MSEs of the proposed SIPC for different numbers of inputs.

A. Computing Accuracy

SIPCs: Fig. 5(a) shows the MSEs of the proposed 9-input unipolar SIPC (P_{LFSR} and P_{Sobol}), which exponentially decrease as the bitwidths of RNSs increase. The unipolar SIPC using an SSG realizes a higher computing accuracy than the one using an LFSR. Fig. 5(b) shows that the proposed 9-input bipolar SIPC achieves much higher computing accuracy than the one sharing RNSs and using circular shifting for decorrelation (S_{LFSR} and S_{Sobol}) [10]. The MSE is lowered by up to 85% and 84% on average compared with S_{LFSR} and S_{Sobol} , respectively. Fig. 6 illustrates the MSEs of the proposed SIPCs for different numbers of inputs, using 8-bit LFSRs and SSGs. As can be seen, the number of inputs affects the computing accuracy of the unipolar SIPCs, and the accuracy loss worsens for the bipolar SIPCs. The MSEs of 1024-input unipolar and 64-input bipolar SIPCs respectively are about 5×10^{-4} and 9×10^{-4} , which are sufficient for image-processing applications because the peak signal-to-noise ratio (PSNR) will not be less than 30 dB [6].

SMCs: Fig. 7 shows the MSEs of the proposed 9-input SMCs (P_{LFSR} and P_{Sobol}), the MUX based designs (M_{LFSR} and M_{Sobol}) [3], the minimum detector based design (D_{LFSR}) [4], and the Two LFSRs based design (T_{LFSR}) [5]. For M_{LFSR} and M_{Sobol} , a 16-input MUX is used, where its first 8 input bitstreams are connected to the first 8 pins, and the other 8 pins are connected to the 9th input bitstream. The 4 select bitstreams that must be uncorrelated with input bitstreams are generated by using isolators, for example, D flip flops (DFFs) [14]. The MUX and detector based SMCs perform better than the proposed designs when the bitwidths of RNSs are less than 6. However, all of them have larger MSEs, which are impractical in reality. As the bitwidths of RNSs increase, the proposed designs consistently outperform the previous designs. For example, the MSEs of the unipolar and bipolar SMCs using 8-bit RNSs are lowered by about 92% and 93%, 87%, and 84%, using LFSRs and SSGs, compared to M_{LFSR} and M_{Sobol} , respectively. The proposed SMCs for different numbers of inputs using 8-bit RNSs bring out the same MSE trend as the SIPC, as shown in Fig. 8.

B. Hardware Cost

SIPCs: TABLE III lists the hardware costs of the proposed 9-input unipolar and bipolar SIPCs, and previous stochastic designs using D flip flops for decorrelation [10], using 8-bit RNSs. All designs include SNGs and the core computing units,

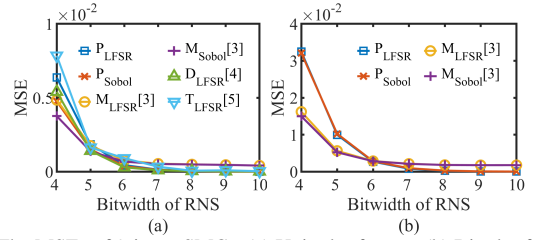


Fig. 7. The MSEs of 9-input SMCs. (a) Unipolar format. (b) Bipolar format.

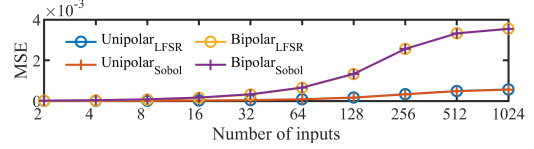


Fig. 8. The MSEs of the proposed SMC for different numbers of inputs.

TABLE III
HARDWARE COSTS OF 9-INPUT SIPCS

Design	Area	Power	Delay	PDP	ADP	EDP	
Unipolar	P_{LFSR}	545.61	19.28	2.38	45.89	1298.54	109.21
	P_{Sobol}	648.98	18.37	2.36	43.35	1531.58	102.31
Bipolar	P_{LFSR}	751.99	19.88	2.56	50.89	1925.10	130.29
	P_{Sobol}	850.60	19.44	2.60	50.54	2211.56	131.41
	S_{LFSR} [10]	1962.63	21.62	2.71	58.59	5318.72	158.78
	S_{Sobol} [10]	2063.53	21.17	2.72	57.58	5612.79	156.62

Area: μm^2 ; Power: μW ; Delay: ns; PDP: $\mu\text{W}\cdot\text{ns}$; ADP: $\mu\text{m}^2\cdot\text{ns}$; EDP: $\mu\text{W}\cdot\text{ns}^2$.

TABLE IV
HARDWARE COSTS OF 9-INPUT SMCs

Design	Area	Power	Delay	PDP	ADP	EDP	
Unipolar	P_{LFSR}	165.64	12.83	0.96	12.32	159.01	11.82
	P_{Sobol}	249.78	12.10	0.96	11.62	239.79	11.15
	M_{LFSR} [3]	318.23	39.10	0.98	38.32	311.86	37.55
	M_{Sobol} [3]	400.60	34.59	0.98	33.90	392.59	33.22
	D_{LFSR} [4]	231.08	21.90	1.56	34.16	360.49	53.30
	T_{LFSR} [5]	235.32	24.22	1.04	25.19	244.73	26.20
Bipolar	P_{LFSR}	165.64	12.84	0.96	12.33	159.01	11.83
	P_{Sobol}	249.78	12.10	0.96	11.62	239.79	11.15
	M_{LFSR} [3]	305.35	40.38	0.98	39.57	299.24	38.78
	M_{Sobol} [3]	388.61	34.48	0.98	33.79	380.84	33.11

Area: μm^2 ; Power: μW ; Delay: ns; PDP: $\mu\text{W}\cdot\text{ns}$; ADP: $\mu\text{m}^2\cdot\text{ns}$; EDP: $\mu\text{W}\cdot\text{ns}^2$.

without the derandomization units for subsequent usage. The proposed SIPC has only a few logic gates on the critical path to realize a smaller delay than S_{LFSR} and S_{Sobol} with multilevel cascaded MUXs. The proposed designs are more efficient than S_{LFSR} and S_{Sobol} in all aspects. For example, the power-delay product (PDP), area-delay product (ADP), and energy-delay product (EDP) are respectively reduced by about 13%, 62%, and 17% for the bipolar SIPCs.

SMCs: TABLE IV gives the hardware costs of the proposed 9-input unipolar and bipolar SMCs, the MUX based designs [3], the minimum detector based design [4], and the two LFSRs based design [5], using 8-bit RNSs. D_{LFSR} uses a multilevel minimum detector to find the smallest value in 9 inputs, which inevitably induces a larger delay. The proposed designs have a decisive advantage over the previous ones. For example, the EDP of the proposed bipolar SMC using an LFSR is reduced by up to about 69%, compared to that of bipolar M_{LFSR} [3].

C. Applications

Five images downloaded from [15], including clock, airplane, cameraman, Lena, and moon, are processed using the SIPCs and SMCs with 8-bit LFSRs, as examples. PSNR and mean structural similarity (MSSIM) are used to evaluate the quality of processed images [16]. Fig. 9 gives the 3×3 kernels for Gaussian filtering, Sobel edge detection, and mean filtering,

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \times \frac{1}{8} \times \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \times \frac{1}{8} \times \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \times \frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Fig. 9. Kernels for three image processing algorithms. (a) Gaussian filtering. (b) Sobel edge detection. (c) Mean filtering.

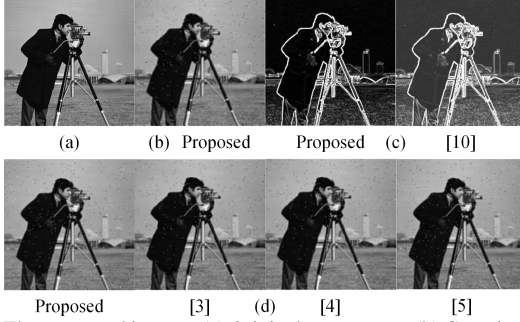


Fig. 10. The processed images. (a) Original cameraman. (b) Gaussian filtering. (c) Sobel edge detection. (d) Mean filtering.

TABLE V
PSNR OF THREE IMAGE PROCESSING ALGORITHMS

Design	Gaussian filtering		Edge detection		Mean Filtering		
	Proposed	Proposed [10]	Proposed [10]	Proposed [3]	[3]	[4]	[5]
PSNR	49.53	35.41	24.58	48.73	35.69	34.39	41.97

TABLE VI
MSSIM OF THREE IMAGE PROCESSING ALGORITHMS

Design	Gaussian filtering		Edge detection		Mean Filtering		
	Proposed	Proposed [10]	Proposed [10]	Proposed [3]	[3]	[4]	[5]
MSSIM	0.9974	0.9887	0.8598	0.9928	0.9867	0.9400	0.9786

where $1/16$, $1/8$, $1/8$, and $1/9$ are factors to scale down inputs into $[0, 1]$ or $[-1, 1]$.

Gaussian filtering: Since the kernel of Gaussian filtering contains only positive numbers, a proposed unipolar SIPC can realize the filtering. In addition, all images are polluted through salt & pepper noise with a density of 0.01 at first using the command ‘imnoise’ in MATLAB and then processed.

Edge detection: The Sobel operators have negative and positive numbers, so a bipolar SIPC is used to perform the edge detection [17], for example, the proposed bipolar SIPC and a previous design in [10]. The involved absolute function is implemented by a finite state machine with 16 states [18].

Mean filtering: Mean filtering is just to average neighboring pixels to replace the central pixel. Thus, both unipolar SMC and bipolar SMC can be used to realize the filtering [3-5]. All images are also polluted using salt & pepper noise.

TABLE V and TABLE VI list the average values of PSNR and MSSIM for five processed images. Fig. 10 shows the original and processed images for these three applications, including cameraman, as an example. The proposed designs provide better results and image quality for all three image processing algorithms.

VI. CONCLUSION

This paper proposes a design method for stochastic inner-product circuits (SIPCs) and stochastic mean circuits (SMCs) using positively correlated bitstreams in the unipolar and bipolar formats, respectively. The SIPC or SMC uses only one random number source (RNS), which significantly reduces hardware costs and increases computing accuracy, compared to previous works. Gaussian filtering, edge detection, and mean filtering validate the strengths of the proposed designs.

REFERENCES

- [1] Y. Zhang, L. Xie, J. Han, X. Cheng, and G. Xie, “Highly accurate and energy efficient binary-stochastic multipliers for fault-tolerant applications,” *IEEE Trans Circuits Syst II Exp Briefs*, vol. 70, no. 2, pp. 771-775, Sep. 2023.
- [2] Y. Liu, S. Liu, Y. Wang, F. Lombardi, and J. Han, “A survey of stochastic computing neural networks for machine learning applications,” *IEEE Trans Neural Netw Learn Syst*, vol. 32, no. 7, pp. 2809-2824, Jul. 2021.
- [3] M. Najafi, and M. Salehi, “A fast fault-tolerant architecture for Sauvola local image thresholding algorithm using stochastic computing,” *IEEE Trans Very Large Scale Integr VLSI Syst*, vol. 24, no. 2, pp. 808-812, Feb. 2016.
- [4] S. Wang, G. Xie, W. Xu, X. Cheng, and Y. Zhang, “High-accuracy mean circuits design by manipulating correlation for stochastic computing,” *Int J Circuit Theory Appl*, vol. 50, no. 10, pp. 3692-3703, Jun. 2022.
- [5] F. Li, G. Xie, J. Han, and Y. Zhang, “Mean circuit design using correlated random bitstreams in stochastic computing,” in 2022 22nd International Conference on Nanotechnology (NANO), Palma de Mallorca, Spain, 2022, pp. 4-7.
- [6] S. Salehi, “Low-cost stochastic number generators for stochastic computing,” *IEEE Trans Very Large Scale Integr VLSI Syst*, vol. 28, no. 4, pp. 992-1001, Apr. 2020.
- [7] 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.
- [8] 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.
- [9] Y. Chang, and K. Parhi, “Architectures for digital filters using stochastic computing,” in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 2013, pp. 2697-2701.
- [10] H. Ichihara, T. Sugino, S. Ishii, T. Iwagaki, and T. Inoue, “Compact and accurate digital filters based on stochastic computing,” *IEEE Trans Emerging Top Comput*, vol. 7, no. 1, pp. 31-43, Mar. 2019.
- [11] 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.
- [12] R. Wang, J. Han, B. Cockburn, and D. Elliott, “Stochastic circuit design and performance evaluation of vector quantization for different error measures,” *IEEE Trans Very Large Scale Integr VLSI Syst*, vol. 24, no. 10, pp. 3169-3183, Oct. 2016.
- [13] “TCBN40LPBWP TSMC N40LP standard cell library datasheet,” Taiwan Semiconductor Manufacturing Company.
- [14] P. Ting, and J. Hayes, “Isolation-based decorrelation of stochastic circuits,” in 2016 IEEE 34th International Conference on Computer Design (ICCD), Scottsdale, AZ, USA, 2016, pp. 88-95.
- [15] “The USC-SIPI image database,” 1981; <https://sipi.usc.edu/database/>.
- [16] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans Image Process*, vol. 13, no. 4, pp. 600-612, Apr. 2004.
- [17] H. Joe, and Y. Kim, “Novel stochastic computing for energy-efficient image processors,” *Electronics*, vol. 8, no. 6, pp. 1-11, Jun. 2019.
- [18] P. Li, D. Lilja, W. Qian, M. Riedel, and K. Bazargan, “Logical computation on stochastic bit streams with linear finite-state machines,” *IEEE Trans Comput*, vol. 63, no. 6, pp. 1473-1485, Jun. 2014.