



**BRIEF REPORT**

# A stochastic computing architecture for local contrast and mean image thresholding algorithm

Wenbing Xu<sup>1</sup>  | Guangjun Xie<sup>1</sup> | Shaowei Wang<sup>1</sup> | Zhendong Lin<sup>1</sup> | Jie Han<sup>2</sup> | Yongqiang Zhang<sup>1</sup> 

<sup>1</sup>School of Microelectronics, Hefei University of Technology, Hefei, China

<sup>2</sup>Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta, Canada

**Correspondence**

Yongqiang Zhang, School of Microelectronics, Hefei University of Technology, Hefei 230601, China.  
Email: [ahzhangyq@hfut.edu.cn](mailto:ahzhangyq@hfut.edu.cn)

**Funding information**

Natural Sciences and Engineering Research Council (NSERC) of Canada, Grant/Award Number: RES0048688; Fundamental Research Funds for the Central Universities of China, Grant/Award Numbers: JZ2020HGQA0162, JZ2020HGTA0085

**Abstract**

Image binarization algorithms in document image analysis divide pixel values into two groups, including white as background and black as foreground. Among others, the local contrast and mean (LCM)-based thresholding algorithm offers excellent performance in processing degraded documents. This algorithm, however, is susceptible to noise and requires significant hardware resources. In this paper, an energy-efficient and fault-tolerant architecture is proposed for implementing the LCM algorithm in stochastic computing (SC). Leveraging correlated input bitstreams, this architecture saves energy and improves the fault tolerance of the implementation. Experimental results show that the proposed LCM stochastic architecture significantly outperforms the stochastic implementation of the Sauvola algorithm in terms of both binarization accuracy and hardware overhead and energy consumption. Even using 16-bit streams, the proposed circuit produces an error rate lower than 5%. The stochastic implementation of the LCM algorithm using a 16-bit length FSM-based LD sequence is 22 times less in area, 26 times less in total power, 28 times less in energy consumption and more fault-tolerant than the conventional 8-bit bit-width weighted binary with the same frequency constraints.

**KEYWORDS**

energy efficiency, fault tolerance, LCM algorithm, stochastic computing

## 1 | INTRODUCTION

Document image binarization is performed during the preprocessing stage of document analysis to segment foreground text from the background. Therefore, a fast and accurate image binarization algorithm is essential for document image processing. Binarization is the process of converting a grayscale image into only two types of pixels, white as background and black as foreground.<sup>1</sup> Binarization algorithms can be divided into two categories, including global and local binarization methods. Global methods attempt to find a threshold for the entire document, such as Otsu.<sup>2</sup> For high-quality document images, the global threshold can effectively extract text rapidly. However, it is not suitable for processing complex or degraded documents. If the illumination on a document is not uniform, global binarization methods tend to produce edge noises along its boundaries. To overcome such complexities, local thresholding methods have been proposed for image binarization. These algorithms estimate a threshold for each pixel based on its neighboring information, such as Sauvola.<sup>3</sup> Local methods can produce acceptable results even for heavily degraded documents, although they are usually slow in processing speed compared with the global ones.

A thresholding algorithm typically depends on a local window size that determines the processing time and computed results. Using the local contrast and mean (LCM), such a method provides excellent results for a large character size even at a small window size.<sup>[4]</sup> With nanoscale CMOS devices, it is expected that circuits to implement these algorithms are more sensitive to environmental noise, process, voltage, and thermal variations.<sup>5</sup> In conventional weighted binary circuit design, fault-tolerant techniques, such as triple modular redundancy, can improve the reliability of the implementation of these algorithms. However, it consumes more hardware and thus more energy, which limits the advantages of these algorithms.

In this paper, an architecture for implementing the LCM algorithm in SC is proposed that leverages the capabilities of SC in reducing circuit complexity and tolerating soft errors, compared with the weighted binary method. The main advantages of the proposed stochastic architecture are summarized as follows:

1. The problem of parameter selection for the stochastic implementation of the LCM algorithm is elaborated in detail, whereas it is not specifically described in Najafi and Salehi.<sup>6</sup> Meanwhile, the proposed stochastic comparator is more energy efficient than in Najafi and Salehi.<sup>6</sup>
2. It produces an acceptable error rate of lower than 5% even when 16-bit streams are used, saving energy and speeding up the processing by exploiting correlated bitstreams.
3. The proposed LCM algorithm stochastic architecture has a smaller area, lower power, and energy consumption and higher fault tolerance than the conventional weighted binary design when using streams lower than 128 bits.

This paper proceeds as follows. Section 2 introduces the concepts of the LCM algorithm and SC. Section 3 presents the parameter selection of the stochastic implementation of the LCM algorithm, the conventional weighted binary implementation, and the proposed stochastic architecture. Section 4 illustrates experimental results. Section 5 concludes this paper.

## 2 | BACKGROUND

### 2.1 | LCM-based thresholding algorithm

The binarization algorithm proposed in Singh and Sinam<sup>4</sup> is performed by a local threshold obtained from LCM values. The threshold  $T(x,y)$  is given by

$$T(x,y) = k \times [m(x,y) + (I_{\max} - I_{\min})(1 - I(x,y))], \quad (1)$$

where  $k \in [0,1]$  is a bias constant,  $m(x,y)$  is the local mean pixel value,  $I_{\max}$  and  $I_{\min}$  are the local maximum and minimum pixel values within a local window size of  $W \times W$ , and  $I(x,y)$  is a center pixel value.

The main function of the bias constant  $k$  is to control the degree of binarization by varying the threshold value. The larger the value of  $k$  is, the larger the threshold value is, and thus a larger number of foreground pixel points will be obtained. If the pixel values in a local window only slightly fluctuate, that is,  $I_{\max} - I_{\min} \approx 0$ , then  $T(x,y) \approx k \times m(x,y)$ . Therefore, the threshold  $T$  depends on the value of the bias constant  $k$ .

### 2.2 | Stochastic computing

SC is an unconventional method of computation that treats data as probabilities, which are encoded as random binary bitstreams.<sup>7</sup> Compared with deterministic weighted binary techniques, SC has the advantage of reducing hardware overhead and enhancing fault tolerance. Thus, it has been widely investigated for computation-intensive applications, such as image processing.<sup>8</sup> Stochastic bitstreams are often referred to as stochastic numbers (SNs) and typically use either unipolar or bipolar encodings. Under the same bitstream length, the accuracy of the unipolar format is twice that of a bipolar one,<sup>7</sup> so the unipolar representation is utilized in this paper.

Some basic arithmetic operations in SC can be simply implemented with combinatorial logic. For example, scaled addition can be implemented by an MUX (Figure 1A), an AND gate (Figure 1E) can realize multiplication, and a NOT gate (Figure 1F) can implement the function of an inverter. Managing and manipulating the correlation between SNs









Operation	(A) Addition	(B) Max	(C) Min	(D) Subtract	(E) Multiply	(F) Inverse	(G) D/S convert	(H) S/D convert
Circuit								
Function	$P_z=0.5(p_x+p_y)$	$P_z=\max(p_x,p_y)$	$P_z=\min(p_x,p_y)$	$P_z= p_x-p_y $	$P_z=p_x \times p_y$	$P_z=1-p_x$	$P_z=B/2^k$	$B=p_x \times 2^k$
Correlation	Uncorrelated	Positive	Positive	Positive	Uncorrelated	NA	NA	NA

FIGURE 1 SC elements. (A) Addition. (B) Max. (C) Min. (D) Subtract. (E) Multiply. (f) Inverse. (G) D/S convert. (H) S/D convert

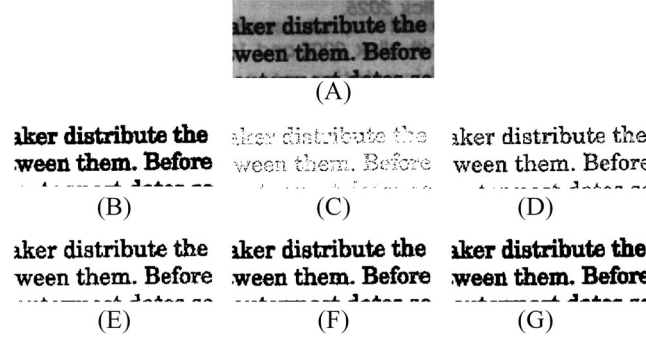


FIGURE 2 The binarization effect of LCM algorithm for different values of  $k$ . (A) Original degraded document image. (B) The binarization benchmark image. (C)  $k = 0.1$ . (D)  $k = 0.2$ . (E)  $k = 0.3$ . (F)  $k = 0.4$ . (G)  $k = 0.5$

in SC is critical for generating accurate results.<sup>9</sup> If two input SNs  $X$  and  $Y$  are maximally positively correlated, OR gates (Figure 1B) and AND gates (Figure 1C) can, respectively, be used to implement max/min functions, and an XOR gate (Figure 1D) performs the subtraction. SC uses a digital-to-stochastic (D/S) converter (Figure 1G) to convert a given binary number  $B \in [0, N]$  to a stochastic bitstream. The converter generates a random integer value  $R \in [0, N]$  by a random number generator (RNG). The value  $R$  is compared with  $B$  to generate an  $N$ -bit SN  $Z$  with  $P_z = B/N$ . To convert an SN back to its binary encoded format, a stochastic-to-digital (S/D) converter (Figure 1H) is used, which consists of a counter that sums up each bit in the SN. The conversion circuits can consume up to 80% of the total area in an SC circuit.<sup>10</sup> Therefore, this paper considers the use of the shared RNG to reduce the hardware cost of the stochastic circuit without compromising its accuracy.

### 3 | IMPLEMENTATIONS OF THE LCM ALGORITHM

#### 3.1 | Thresholding algorithm parameters

Considering (1), the performance of the LCM algorithm depends on two parameters: (1) a bias constant  $k$  and (2) a selected window size  $W \times W$ . Assuming that the value of  $k$  takes 1, the maximum value of Equation 1 is 2. To implement this algorithm by leveraging SC, all image pixel values have to be scaled down from  $[0, 255]$  to  $[0, 1]$ . Therefore, the range of  $k \in [0, 0.5]$ . Assuming that the selected window size is  $5 \times 5$ , the images are binarized with  $k = 0.1, 0.2, 0.3, 0.4, 0.5$ , and the mean absolute error (MAE) and peak signal-to-noise ratio (PSNR) are used as accuracy evaluation to select the appropriate values of  $k$ . The PSNR is given as<sup>11</sup>

$$PSNR = 20 \log(MAX_I) - 10 \log(MSE), \quad (2)$$

where  $MAX_I$  is defined as the max intensity value. Figure 2 shows the binarization effect of the LCM algorithm for different values of  $k$ . Based on our experiments on several image datasets, Figure 2B was selected as the benchmark image for binarization. Table 1 shows the MAE and PSNR of image binarization for different values of  $k$ . It can be seen that the MAE is the lowest and the PSNR is the maximum when  $k = 0.5$ . This means that the larger the value of  $k$ , the larger the threshold value. The larger the threshold, the more foreground pixels are obtained, so  $k$  being defined as 0.5 is an appropriate value.

TABLE 1 The MAE(%) and PSNR (dB) of image binarization at different values of  $k$ 

$k$	0.1	0.2	0.3	0.4	0.5
MAE	$1.90 \times 10^{-1}$	$1.06 \times 10^{-1}$	$6.44 \times 10^{-2}$	$3.62 \times 10^{-2}$	$1.29 \times 10^{-2}$
PSNR	7.21	9.76	11.91	14.41	<b>18.88</b>

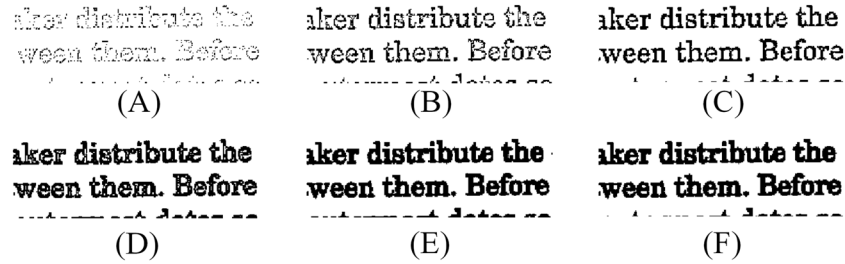
FIGURE 3 (A)  $3 \times 3$ , (B)  $5 \times 5$ , and (C)  $7 \times 7$  window size using the Sauvola algorithm. (D)  $3 \times 3$ , (E)  $5 \times 5$ , and (F)  $7 \times 7$  window size using LCM algorithm

TABLE 2 The PSNR (dB) of LCM and Sauvola algorithms for binarization of images with different window sizes

Window size	$3 \times 3$	$5 \times 5$	$7 \times 7$	$9 \times 9$	$11 \times 11$	$13 \times 13$
LCM algorithm	14.10	<b>18.88</b>	18.76	18.34	17.27	16.04
Sauvola algorithm	7.07	9.19	10.61	11.06	11.37	11.67

The quality of binary images processed by the LCM algorithm also depends on the selected window size, while an oversized window would not significantly improve the quality, even at the cost of additional hardware resources.<sup>4</sup> To verify the effect of window size on the quality of binarized images and determine a suitable size, we compare the LCM algorithm under the deterministic weighted binary technique with the Sauvola algorithm in Najafi and Salehi.<sup>6</sup> Figure 2A shows the original degraded document image. Figure 3A–C and Figure 3D–F are, respectively, the images processed using Sauvola and LCM algorithms with  $3 \times 3$ ,  $5 \times 5$ , and  $7 \times 7$  window size. It can be seen that the LCM algorithm produces a better-binarized image than the Sauvola algorithm. For example, when the window size is  $3 \times 3$ , the binarized image using the Sauvola algorithm is unclear and many foreground pixels are missing.

Table 2 shows the PSNR of LCM and Sauvola algorithms for the binarization of images with different window sizes. It can be seen that the PSNR of the LCM algorithm is higher than that of the Sauvola algorithm, regardless of the window size. And when the window size is  $5 \times 5$ , the PSNR of the LCM algorithm is the highest. Meanwhile, we also count the background pixels of the images processed by different window sizes in Figure 4. Compared with the image processed by the LCM algorithm at a  $13 \times 13$  window size, the average error of the image processed at a  $5 \times 5$  window size is only 1.8%, which is acceptable for an error lower than 2%.<sup>12</sup> In summary, to balance the hardware overhead and time consumption, a window size of  $5 \times 5$  is chosen in this paper.

### 3.2 | Conventional implementation

The key to the LCM algorithm is to calculate the threshold  $T(x,y)$  for each pixel in an image, which involves finding the maximum, minimum, and mean of the pixel values in a local window. For a window size of  $5 \times 5$ , we need to calculate them in this window using 25-pixel values, denoted as  $z_1, z_2, \dots, z_{25}$ . The mean of these pixel values in the  $5 \times 5$  window is then

$$\text{mean} = \frac{z_1 + z_2 + z_3 + \dots + z_{24} + z_{25}}{25}. \quad (3)$$

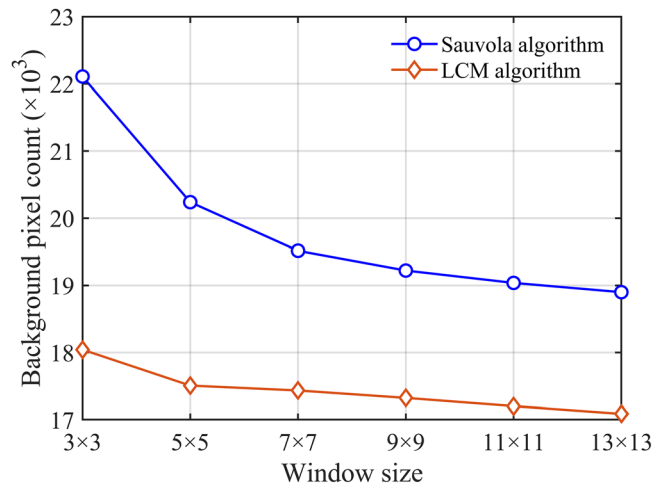


FIGURE 4 Background pixel value statistics for the binarized images using different window sizes [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

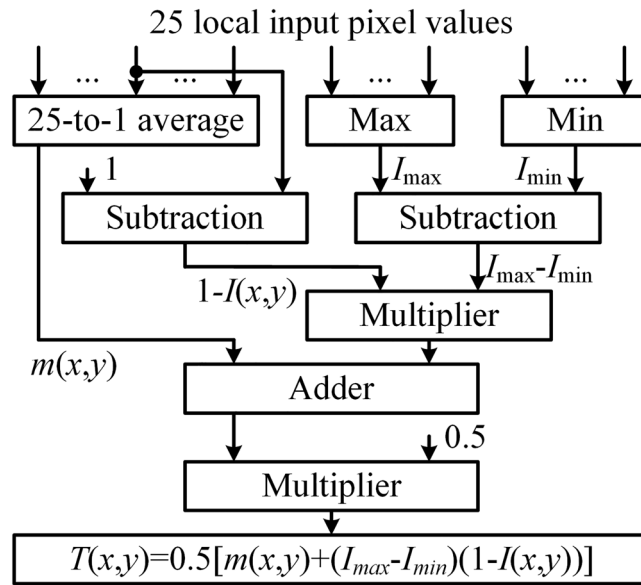


FIGURE 5 Conventional implementation of the LCM algorithm

A five-stage MAX tree is used to find the maximum among 25-pixel values, in which each MAX module evaluates two neighboring values to output a larger one for the next stage until it is found. The critical path delay of the circuit designed in this way is shorter than a simple sequential comparison. The minimum pixel value is found similarly. Figure 5 shows a schematic for the conventional implementation of the LCM algorithm.

### 3.3 | Stochastic implementation

From the analysis in Section 3.1, a new modified threshold  $T(x,y)$  in the LCM algorithm is proposed as

$$T(x,y) = 0.5 \times [m(x,y) + (I_{max} - I_{min})(1 - I(x,y))]. \tag{4}$$

Implementing (3) requires three steps: (1) converting the pixel values of an image into stochastic bitstreams; (2) generating a threshold bitstream by a stochastic circuit; and (3) determining the output binary values.

1. *Converting pixel values into stochastic bitstreams*: The D/S converter in Lee et al.<sup>9</sup> is used to convert pixel values to stochastic bitstreams. An RNG usually uses logic circuits such as linear feedback shift registers (LFSRs) instead of true random sources. Its output is repeatable, but it has many characteristics of a truly random number. In recent years, some low-discrepancy (LD) sequences, such as Halton<sup>13</sup> and Sobol<sup>14</sup> sequences, have been proposed to improve the accuracy of the circuits. In terms of hardware overhead, the generation of Halton sequences requires the use of counters with different base- $b$  when several independent sequences are required, where  $b$  is a prime number. The generation of the Halton sequence must be base converted, which imposes additional hardware overhead on the stochastic circuit. In addition, the hardware overhead of the Sobol is also significant.<sup>14</sup> To solve the issue of costly LD bit-stream generators, a low-cost FSM-based LD bit-stream generator is proposed for SC designs that require multiple independent bitstreams.<sup>15</sup> In this paper, the LFSR and FSM-based LD bitstream generators are used as RNGs to generate stochastic bitstreams. Note that for the input of the MUX, the D/S converter shares an RNG to generate all the required stochastic bitstreams, and the selection signal of the MUX uses the separate RNG to generate the stochastic bit-streams.
2. *Generating threshold bitstreams*: First, an averaging operation should be taken for 25 input bitstreams in a  $5 \times 5$  window. For this purpose, a 5-to-1 stochastic mean circuit (SMC) is proposed by using an 8-1 MUX as shown in Figure 6A, and Figure 6B shows its symbol. The fifth input bit-stream  $z_5$  is copied three times to connect to its last three inputs, and bitstreams  $z_1, z_2, z_3, z_4$  are connected to its first four inputs in order. The most significant bit  $S_3$  of the MUX is connected to a bitstream encoding 0.2, and the remaining bits are connected to bitstreams encoding 0.5. The toggle flip-flop (TFF) is an interesting element in SC, where the output has a probability of 1/2, independent of the probability of its inputs, as long as the inputs are not zero.<sup>16</sup> Thus, the output bitstream of the MUX is the average of five input bitstreams. This has the advantage that the area cost of the TFF is no more than the RNG required to generate 1/2. Then, by combining five separate 5-to-1 SMCs and averaging their outputs by using an additional 5-to-1 SMC, the final bitstream obtained is exactly the average of 25 input bitstreams, as shown in Figure 6C, and Figure 6D shows its symbol.

The accuracy of the proposed 5-to-1 and 25-to-1 SMCs are evaluated. Absolute error (AE) and mean square error (MSE) are used as metrics to quantify the accuracy. We use the *rand* function to select 5 and 25 numbers and then perform 1000 Monte Carlo simulations. Figure 7A,B shows the AE of the 5-to-1 and 25-to-1 SMCs, respectively. The length of the stochastic bitstreams is assumed to be 1024 bits. It can be seen that the AE of the proposed 5-to-1 SMC mostly concentrates below 0.05 accounting for about 96.5%, while that of 25-to-1 SMC concentrates below 0.04 accounting for about 97.3%. Table 3 shows the MSE of the proposed SMC for different stochastic bitstream lengths.

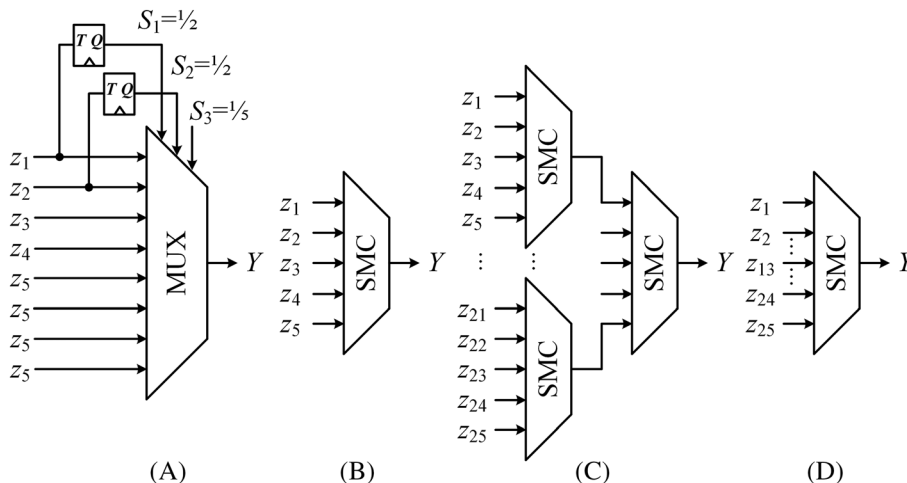


FIGURE 6 The proposed stochastic mean circuit. (A) 5-to-1 SMC schematic. (B) 5-to-1 SMC symbol. (C) 25-to-1 SMC schematic. (D) 25-to-1 SMC symbol

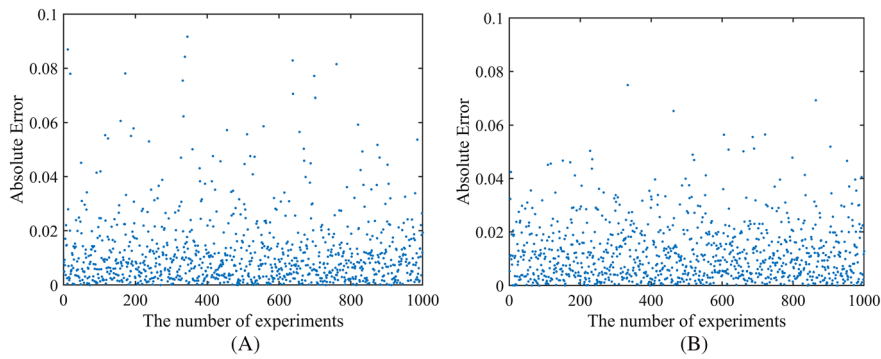


FIGURE 7 The absolute error of the proposed SMC. (A) The AE of 5-to-1 SMC. (B) The AE of 25-to-1 SMC [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

TABLE 3 The MSE of SMC with different stochastic bitstream lengths

Bitstream lengths (bit)	64	128	256	512	1024
5-to-1 SMC	$6.00 \times 10^{-3}$	$3.00 \times 10^{-3}$	$2.30 \times 10^{-3}$	$1.20 \times 10^{-3}$	$4.89 \times 10^{-4}$
25-to-1 SMC	$3.90 \times 10^{-3}$	$2.00 \times 10^{-3}$	$1.10 \times 10^{-3}$	$5.91 \times 10^{-4}$	$2.43 \times 10^{-4}$

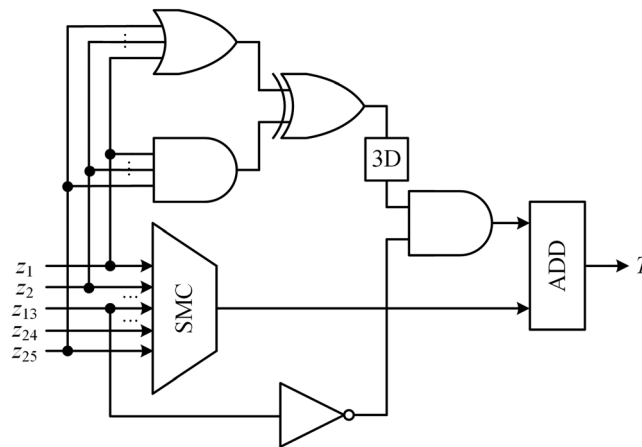


FIGURE 8 The proposed stochastic implementation of the LCM algorithm

Second, the maximum and minimum bit-streams are generated among 25 input bitstreams. A stochastic max/min function circuit based on shift registers is proposed in Lunglmayr et al,<sup>17</sup> while the circuit requires a long shift register and its accuracy depends on the length of the bitstreams. To exactly obtain the maximum  $I_{\max}$  and minimum  $I_{\min}$ , OR/AND gates can be utilized by exerting maximally positively correlated input bitstreams.<sup>9</sup> To obtain the local contrast, it is also necessary to calculate the difference  $I_{\max} - I_{\min}$  between the  $I_{\max}$  bitstream and the  $I_{\min}$  bitstream. Because they are maximally correlated, an XOR gate can perform the subtraction on two bitstreams, resulting in high accuracy and low hardware consumption.

Lastly, several simple operations need to be designed for completing (3). A NOT gate can perform  $1 - I(x,y)$ , where  $I(x,y) = z_{13}$ . Then an AND gate performs the multiplication of  $(I_{\max} - I_{\min})(1 - I(x,y))$ . Because the multiplication realized using an AND gate requires its input bitstreams to be independent of each other, an isolation method for decorrelating them is proposed by inserting D flip-flops (DFFs) following the XOR gate. This isolation method also incurs very low hardware cost and low latency, compared with regeneration.<sup>18</sup> The threshold bit-stream  $T(x,y)$  is calculated by using a scaled addition circuit.<sup>19</sup> Figure 8 shows the proposed stochastic implementation of the LCM algorithm, where “3D” means the delay of three clock cycles to reduce the correlation between input bitstreams.

3. *Determining the binary output values:* This binary value is generated by comparing the produced threshold bit-stream with the pixel bitstream, as

$$b(x,y) = \begin{cases} 0, & \text{if } I(x,y) \leq T(x,y) \\ 1, & \text{otherwise} \end{cases}, \quad (5)$$

where  $b(x,y)$  is the binarized value and  $I(x,y)$  is the pixel value at location  $(x,y)$  in an image. A stochastic comparator is responsible for this part of the proposed circuit.

A stochastic comparator based on a stochastic  $\tanh$  function and scaled subtraction is proposed in Li et al.<sup>8</sup> However, this comparator can only generate ideal results when both input bitstreams are in the bipolar format and their difference is greater than 0.2. To make it simpler and more accurate, a new type of stochastic comparator based on a simple counter is proposed in Najafi and Salehi.<sup>6</sup> This comparator can work in the unipolar format, and its results are also accurate for almost equal input bitstreams. However, this comparator processes bitstream  $B$  after bitstream  $A$  is processed, which causes a relatively large delay. An improved stochastic comparator is proposed, as shown in Figure 9. This comparator uses two  $N$ -bit counters to count the number of 1 s in bitstream  $A$  and bitstream  $B$  at the same time. After that, its output is a 1 if the value of  $A$  is greater than the value of  $B$ ; otherwise, it is a 0. The energy per operation (EPO) is used in Liu and Han<sup>14</sup> to quantify the performance of the circuit. The EPO is given as

$$EPO = Power \times T_{clk} \times L, \quad (6)$$

where  $L$  is the sequence length,  $T_{clk}$  is the clock period, and  $Power$  is the measured power at  $T_{clk}$ . The kernel hardware resources of the stochastic comparator proposed in Najafi and Salehi<sup>6</sup> and the one proposed in this paper are then compared. These two comparators were synthesized using Synopsys Design Compiler with the 40-nm technology library at 100 MHz. The efficiency of the hardware implementation is measured by EPO and runtime. Let  $N = 2^k$  be the length of the bitstreams. The results are shown in Figure 10A–D. It can be seen that although the area and power consumption of the proposed stochastic comparator are higher than those proposed in Najafi and Salehi,<sup>6</sup> the proposed circuit has significant advantages in terms of running time and energy efficiency.

## 4 | EXPERIMENTAL RESULTS

The experimental results of the conventional and the proposed stochastic implementations of the LCM algorithm are verified by using MATLAB and the Verilog HDL languages. The accuracy of a stochastic implementation is determined by the length of its bit-streams used. However, a longer bitstream means that the system consumes higher energy. To evaluate the trade-off between accuracy, hardware cost, and runtime, stochastic circuits with various bitstream lengths (16, 32, 64, 128, and 256) are considered. After being successfully verified on FPGAs, these circuits are then synthesized using the Synopsys Design Compiler NXT R-2020.09-SP5 version with TSMC's 40-nm library, under 100 MHz frequency. The power consumption is measured using Synopsys PrimePower O-2018.06-SP3 version, vector-free power analysis mode. The operating conditions for each implementation are defined by a supply voltage of 1.10 V and a temperature of 25°Celsius. A grayscale degraded document image (Figure 2A) is selected and different implementations are compared, in terms of the hardware cost, energy consumption, and fault tolerance.

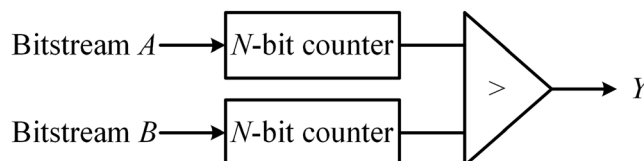


FIGURE 9 The proposed stochastic comparator



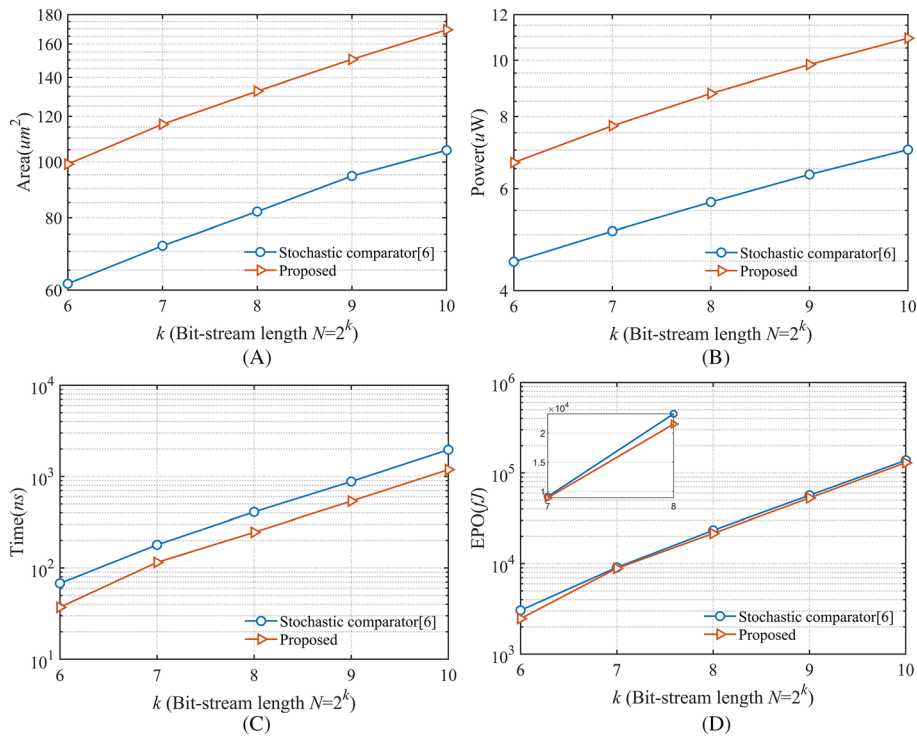


FIGURE 10 Circuit performance comparison. (A) Area comparison. (B) Power comparison. (C) Runtime comparison. (D) EPO comparison [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

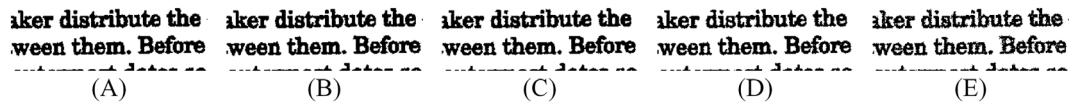


FIGURE 11 Simulation results of the stochastic implementation of the LCM algorithm using LFSR sequences. (A) Stochastic 256. (B) Stochastic 128. (C) Stochastic 64. (D) Stochastic 32. (E) Stochastic 16

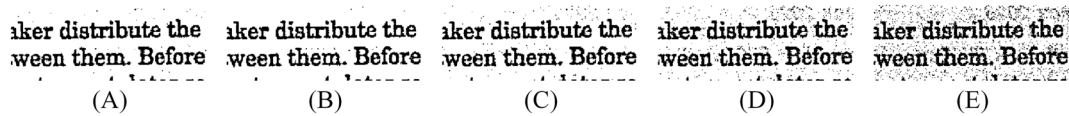


FIGURE 12 Simulation results of the stochastic implementation of the Sauvola algorithm using LFSR sequences in Najafi and Salehi.<sup>[6]</sup> (A) Stochastic 256. (B) Stochastic 128. (C) Stochastic 64. (D) Stochastic 32. (E) Stochastic 16

### 4.1 | Simulation results

Figures 11 and 12 show the simulation results of the LCM algorithm and the stochastic implementation of the Sauvola algorithm in Najafi and Salehi<sup>6</sup> for different bitstream lengths using LFSR sequences. It can be seen that the quality of the stochastic implementation of the LCM algorithm is significantly better than that of the Sauvola algorithm when using bitstream lengths lower than 128 bits. For example, when using 16-bit streams, the binarized image processed by the Sauvola algorithm is full of noise, while the LCM algorithm processes it with better quality. Table 4 shows the MAE and PSNR of the stochastic LCM algorithm circuit with LFSR and FSM-based LD bitstream, respectively, compared with the conventional implementation. It can be seen that the stochastic circuit with 256-bit streams using LFSR sequences produces almost identical results as those produced by the conventional one, with an error rate of 0.52%. Because an MAE of below 5% is acceptable in many digital image processing algorithms,<sup>20</sup> the stochastic implementation of the LFSR sequence using a 16-bit stream is sufficient for the LCM algorithm, which comes with an error rate of

**TABLE 4** The MAE(%) and PSNR (dB) of the stochastic LCM algorithm circuit compared with the conventional implementation

Conventional	Precision		8-bit	7-bit	6-bit	5-bit	4-bit
	MAE		0	2.00	3.55	4.25	4.45
	PSNR		18.88	16.14	14.04	13.34	13.13
SC	Bitstream length		256-bit	128-bit	64-bit	32-bit	16-bit
	LFSR	MAE	0.52	0.63	0.90	1.27	2.14
		PSNR	18.32	17.99	17.42	16.71	15.49
	FSM-based LD	MAE	0.43	0.48	0.56	0.76	1.54
		PSNR	18.61	18.53	18.05	17.83	15.91

**TABLE 5** Circuit performance comparison of conventional and stochastic implementations of the LCM algorithm

Method	Precision	Area ( $\mu\text{m}^2$ )	Power ( $\mu\text{W}$ )	CPD (ns)	Latency (ns)	PLP (pJ)	
Conventional	8-bit	3954.71	369.59	6.82	6.82	2.52	
	6-bit	2939.18	273.51	5.86	5.86	1.60	
	4-bit	1038.29	168.28	3.95	3.95	0.66	
SC	LFSR	256-bit	352.62	30.20	0.63	161.28	4.87
		128-bit	<b>308.17</b>	<b>27.09</b>	<b>0.60</b>	75.52	<b>2.05</b>
		64-bit	<b>272.71</b>	<b>23.87</b>	<b>0.56</b>	35.84	<b>0.85</b>
		32-bit	<b>236.55</b>	<b>20.69</b>	<b>0.52</b>	16.64	<b>0.34</b>
		16-bit	<b>197.74</b>	<b>17.01</b>	<b>0.49</b>	7.84	<b>0.13</b>
	FSM-based LD	256-bit	310.99	22.64	0.65	166.40	3.76
		128-bit	<b>279.59</b>	<b>20.57</b>	<b>0.62</b>	<b>79.36</b>	<b>1.63</b>
		64-bit	<b>237.25</b>	<b>18.36</b>	<b>0.48</b>	<b>30.72</b>	<b>0.56</b>
		32-bit	<b>211.50</b>	<b>16.20</b>	<b>0.46</b>	<b>14.72</b>	<b>0.24</b>
		16-bit	<b>179.04</b>	<b>14.11</b>	<b>0.38</b>	<b>6.08</b>	<b>0.09</b>

**TABLE 6** Circuit performance comparison of conventional and stochastic implementations of the Sauvola algorithm

Method	Precision	Area ( $\mu\text{m}^2$ )	Power ( $\mu\text{W}$ )	CPD (ns)	Latency (ns)	PLP (pJ)
Conventional	8-bit	23607.79	2686.50	8.47	8.47	22.75
SC	256-bit	845.13	51.73	2.10	537.60	27.81
	128-bit	760.98	46.90	1.99	254.72	11.95
	64-bit	669.97	42.11	1.77	113.28	4.77
	32-bit	588.12	36.80	1.59	50.88	1.87
	16-bit	494.45	31.72	1.27	20.32	0.64

2.14%. However, using an FSM-based LD bit-stream results in a lower MAE and higher PSNR for the stochastic circuit because the random fluctuations of the pseudo-random bitstream are eliminated. Meanwhile, the accuracy based on the conventional implementation of 8-bit data bit-width is quantized. As can be seen in Table 4, the MAE of the image binarization circuit becomes higher and the PSNR is considerably lower when being quantized to 6 or 4 bits.

## 4.2 | Circuit performance comparison

Tables 5 and 6 show the circuit performance comparison of conventional and stochastic implementations of LCM and Sauvola algorithms for different bit-stream lengths, respectively. Regardless of the binarization algorithm, the stochastic

implementation has a much lower hardware overhead than the conventional ones, which is an advantage of stochastic computing even if the accuracy of the conventional implementation is quantized. For example, the stochastic circuit using LFSR sequences with a length of 256 bits performs up to  $11 \times$  improvement compared with the area of a conventional circuit implementing the LCM algorithm with an 8-bit data bit-width. It also reaches approximately  $12 \times$  improvement in terms of total power. The hardware overhead is still higher than a stochastic implementation when quantizing the conventional 8-bit data bit-width precision to 4 bits. As can be seen in Table 5, the hardware overhead of using FSM-based LD sequences is lower than that of using LFSR sequences. For example, the FSM-based LD sequence reduces the stochastic implementation by 9.3% and the total power consumption by 24% compared with the LFSR-based sequence when the bit-stream length is 128 bits. Meanwhile, Table 6 shows that the hardware overhead of implementing the Sauvola algorithm using LFSR sequences is much higher than that of the LCM algorithm. Note that in this performance comparison, the total area in the proposed stochastic circuits includes all the D/S and S/D converters.

### 4.3 | Energy consumption comparison

Although a stochastic circuit is smaller, it may consume more energy because of the long bit-streams. If using a length of an  $L$ -bit stream to encode a pixel value, the stochastic circuit will be  $L$  times slower than its conventional one at the same operating frequency.<sup>8</sup> The power–latency product ( $PLP$ ) is then used to evaluate the energy consumption, as in

$$PLP = Power \times Latency. \quad (7)$$

The circuit latency of various implementations of the LCM algorithm is shown in Table 5. Note that the “Latency” refers to the product of the critical path delay (CPD) of a circuit and its bitstream length  $L$ . The stochastic implementation using the LFSR sequence consumes less energy than the conventional one when  $L \leq 128$ , regardless of whether the conventional 8-bit bit-width implementation quantizes to 6 or 4 bits, the same is true using FSM-based LD sequences. However, a stochastic implementation using FSM-based LD sequences will consume less energy than using LFSR sequences. For example, a conventional implementation with a 4-bit width consumes about seven times more energy than a stochastic circuit using an FSM-based LD sequence of length 16 bits. As can be seen in Table 6, a hardware implementation of the LCM algorithm would be more energy efficient. For example, a stochastic implementation of the Sauvola algorithm using LFSR sequences consumes approximately six times more energy than an LCM algorithm using LFSR sequences when using 128-bit streams.

### 4.4 | Fault-tolerance comparison

Similar to the method used in Qian et al,<sup>10</sup> various fault rates are injected into the aforementioned circuits. That is, it is simulated by independently flipping a given percentage of bits in the input bitstreams in a circuit. For example, a soft

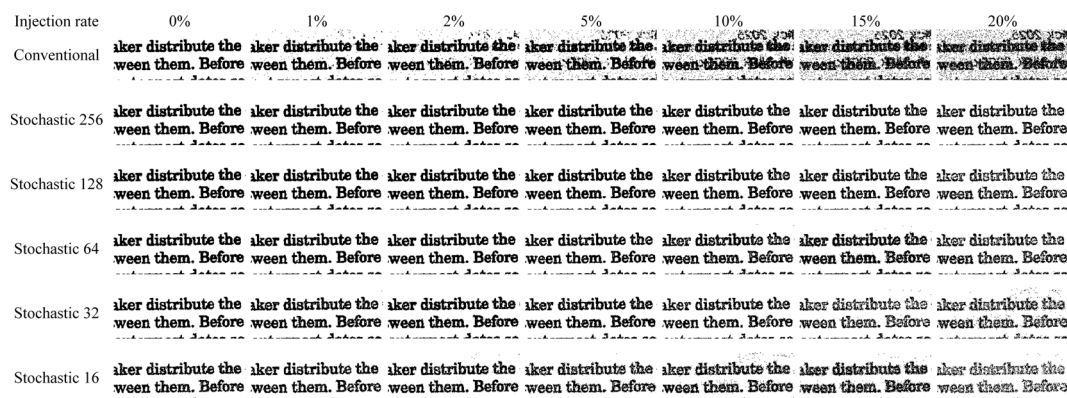
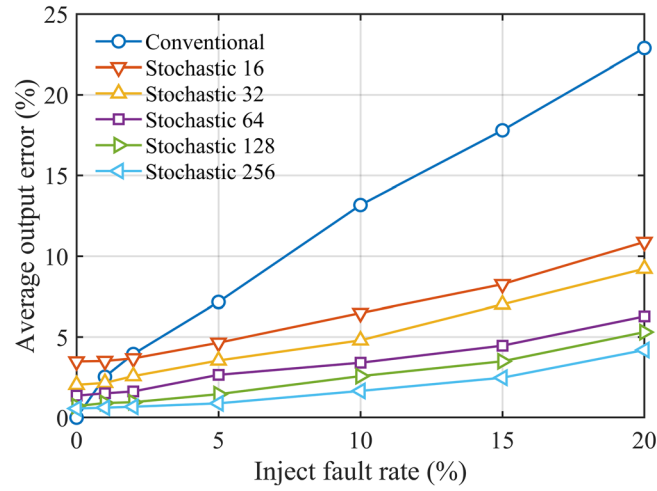


FIGURE 13 The images processed by using conventional and the proposed stochastic implementations for the LCM algorithm versus injecting fault rate



**FIGURE 14** Average output error versus inject fault rate for the conventional and the proposed stochastic implementations of the LCM algorithm [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

error rate of 8% means that 8% of the total number of bits is randomly chosen and flipped. Figure 13 shows the images processed by the conventional and various stochastic implementations of the LCM algorithm versus various inject fault errors using the FSM-based LD sequence. The average output error is then measured. If the height and width of the image are defined as  $H$  and  $W$ , the average output error  $E$  is given by

$$E = \frac{\sum_{i=1}^H \sum_{j=1}^W |T_{ij} - S_{ij}|}{255 \cdot H \cdot W} \times 100\%, \quad (8)$$

where  $S_{ij}$  is an output image pixel processed by the conventional circuit without any injected noise and  $T_{ij}$  is an output image pixel processed by a circuit with injected noise. Figure 14 shows the average output error versus inject fault rate for the conventional and the proposed stochastic implementations of the LCM algorithm. When the injection noise rate is larger than 1%, the stochastic circuit has higher fault tolerance than the conventional one except for the one using 16-bit streams. When the injection noise rate is larger than 2%, even the stochastic circuit using 16-bit streams outperforms the conventional one.

## 5 | CONCLUSION

In this paper, an energy-efficient and fault-tolerant stochastic computing architecture are proposed for implementing the LCM algorithm to binarize degraded document images. To implement this algorithm, a novel stochastic comparator and mean circuit are proposed by employing the correlation between bit-streams to calculate the maximum and minimum pixel values. By balancing the runtime and accuracy of binarized results, an acceptable window size of  $5 \times 5$  is chosen. Experimental results show that the stochastic implementation of the LCM algorithm using FSM-based LD sequences significantly outperforms the conventional implementation in terms of hardware overhead, energy consumption, and fault tolerance. Meanwhile, the proposed LCM algorithm stochastic architecture is significantly better than the Sauvola algorithm in terms of binarization accuracy, hardware overhead, and energy consumption.

## ACKNOWLEDGMENT

This work was supported by the Fundamental Research Funds for the Central Universities of China under Grant No. JZ2020HGTA0085 and No. JZ2020HGQA0162, and the Natural Sciences and Engineering Research Council (NSERC) of Canada (Project Number: RES0048688).

## DATA AVAILABILITY STATEMENT

Research data are not shared.

## ORCID

Wenbing Xu  <https://orcid.org/0000-0002-5796-1406>

Yongqiang Zhang  <https://orcid.org/0000-0003-1403-9128>

## REFERENCES

1. Stathis P, Kavallieratou E, Papamarkos N. An evaluation technique for binarization algorithms. *J Univ Comput Sci.* 2008;14(18):3011-3030.
2. Otsu N. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics.* 1979;9(1):62-66. doi:10.1109/TSMC.1979.4310076
3. Sauvola J, Pietikäinen MJPR. Adaptive document image binarization. 2000;33(2):225-236. doi:10.1016/S0031-3203(99)00055-2
4. Singh OI, Sinam T. Local contrast and mean based thresholding technique in image binarization. *International Journal of Computer Applications.* 2012;51:4-10.
5. Han J, Chen H, Liang JH, Zhu PC, Yang ZX, Lombardi F. A stochastic computational approach for accurate and efficient reliability evaluation. *IEEE Transactions on Computers.* 2014;63(6):1336-1350. doi:10.1109/TC.2012.276
6. Najafi MH, Salehi ME. A fast fault-tolerant architecture for Sauvola local image thresholding algorithm using stochastic computing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems.* 2016;24(2):808-812. doi:10.1109/TVLSI.2015.2415932
7. Alaghi A, Hayes JP. Survey of stochastic computing. *Acm Transactions on Embedded Computing Systems.* 2013;12(2s):1-19. doi:10.1145/2465787.2465794
8. Li P, Lilja DJ, Qian WK, Bazargan K, Riedel MD. Computation on stochastic bit streams digital image processing case studies. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems.* 2014;22(3):449-462. doi:10.1109/TVLSI.2013.2247429
9. Lee VT, Alaghi A, and Ceze L. Correlation manipulating circuits for stochastic computing. In: 2018 Design, Automation & Test in Europe Conference & Exhibition. 2018;1417-1422.
10. Qian WK, Li X, Riedel MD, Bazargan K, Lilja DJ. An architecture for fault-tolerant computation with stochastic logic. *IEEE Trans Comput.* 2011;60(1):93-105. doi:10.1109/TC.2010.202
11. Priyadarshni M, Raj Gupta A, Nithish Kumar V, Kumaravel S. An error efficient and low complexity approximate multi-bit adder for image processing applications. *International Journal of Circuit Theory and Applications.* 2021;49(8):2373-2381. doi:10.1002/cta.3074
12. Alaghi A. The logic of random pulses: stochastic computing. Ph.D., University of Michigan, Ann Arbor, 2015.
13. Alaghi A, Hayes JP, and IEEE. Fast and accurate computation using stochastic circuits. In: Design Automation and Test in Europe Conference and Exhibition, 2014.
14. Liu ST, Han J. Toward energy-efficient stochastic circuits using parallel Sobol sequences. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems.* 2018;26(7):1326-1339. doi:10.1109/TVLSI.2018.2812214
15. Asadi S, Najafi MH, and Imani M. A low-cost FSM-based bit-stream generator for low-discrepancy stochastic computing. In: 2021 Design, Automation & Test in Europe Conference & Exhibition. 2021;908-913.
16. Lee VT, Alaghi A, Hayes JP, Sathe V, Ceze L, and IEEE. Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing. In: Design Automation and Test in Europe Conference and Exhibition. 2017;13-18.
17. Lunglmayr M, Wiesinger D, Haselmayr W. Design and analysis of efficient maximum/minimum circuits for stochastic computing. *IEEE Trans Comput.* 2019;1-1.
18. Ting P and Hayes JP. Isolation-based decorrelation of stochastic circuits. In: 2016 IEEE 34th International Conference on Computer Design (ICCD). 2016;88-95.
19. Lee VT, Alaghi A, Hayes JP, Sathe V, and Ceze L. Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing. In: Design, Automation & Test in Europe Conference & Exhibition. 2017;13-18.
20. Li P, Lilja DJ, and IEEE. Accelerating the performance of stochastic encoding-based computations by sharing bits in consecutive bit streams. In: IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors (ASAP), Washington, DC. 2013;257-260.

**How to cite this article:** Xu W, Xie G, Wang S, Lin Z, Han J, Zhang Y. A stochastic computing architecture for local contrast and mean image thresholding algorithm. *Int J Circ Theor Appl.* 2022;1-13. doi:10.1002/cta.3320