Highly Accurate Division and Square Root Circuits by Exploiting Signal Correlation in Stochastic Computing

Shaowei Wang[1], Guangjun Xie[1], Jie Han[2], and Yongqiang Zhang [1]

Summary

Stochastic computing (SC) is an approximate computing paradigm using probabilities and aims at realizing circuits with low hardware cost. Basic operations (such as addition) have been comprehensively studied, whereas there are few studies on nonlinear operations (such as division and square root) in SC. In this paper, a stochastic division circuit is proposed by using maximally correlated input bitstreams to eliminate the necessity for distinguishing the divisor and dividend. Additionally, four stochastic square root circuits are designed with improved accuracy by decreasing the correlation between intermediate bitstreams via inserting delay elements. Experimental results show that both the proposed division and square root circuits achieve lower mean squared errors (MSEs) while requiring nearly the same hardware resources, compared to the state-of-the-art designs. This result shows the potential in exploiting signal correlation in SC circuit design for high accuracy.

KEYWORDS

Stochastic computing, division circuit, square root circuit, correlation

1. INTRODUCTION

Stochastic computing (SC) is a probabilistic computing paradigm using conventional digital elements [1, 2]. A real number in SC is encoded in the probability of 1s appearing in a random or pseudorandom bitstream, called a stochastic number (SN) [3, 4]. For example, a real number x is represented by an SN X, where x=P(X) meaning the probability of 1s in X, and $0 \le x \le 1$. SC belongs to approximate computing paradigm which can reduce the hardware cost of the circuit [5, 6]. Converting real numbers into SNs requires stochastic number generators (SNGs). Generally, an SNG consists of two components, a random number generator (RNG) and a comparator (CMP). A common RNG uses the linear feedback shift register (LFSR), which is a pseudo-random number source with a simple structure. An SNG composed of an N-bit LFSR generates SNs with a length of (2^N-1) bits in (2^N-1) clock cycles, except for all 0 states.

SC uses simple logic elements to implement basic arithmetic functions. For example, a multiplexer (MUX) realizes the function of mixing two bitstreams to serve as a scaled adder when the select signal is set to ½. Its output is P(Z)=½(P(X)+P(Y)), where X and Y are the input bitstreams, and Z is the output bitstream [7]. A reconfigurable stochastic architecture is applied to synthesize functions by using Bernstein polynomials [8]. The functions such as trigonometric, exponential, logarithmic and sigmoid, can be implemented in SC by using Maclaurin series expansion or factorization [9]. To implement nonlinear functions, sequential logic is generally required and the linear finite-state machine (FSM) has been used to implement complex functions [10]. The division and square root operations were first studied by Gains [11]. Recently, a so-called correlated division (CORDIV) employs the MUX and bitstream correlations [12]. A divider using the saturating subtractor (SSDIV) has been designed by using JK flip-flops (JKFFs) and leveraging correlation [13]. Although some square root circuits were investigated in [14], the related work is still very rare.

To realize highly accurate nonlinear operations, division and square root circuits are investigated by exploiting signal correlation. The main contributions of this work include: 1. A divider design by using the maximally correlated input bitstreams to eliminate the necessity for distinguishing divisor and dividend beforehand. 2. Four square root circuits designed by using

[1]School of Microelectronics, Hefei University of Technology, Hefei, China
[2]Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada
Correspondence
Yongqiang Zhang, School of Microelectronics, Hefei University of Technology, Hefei 230601, China
Email: ahzhangyq@hfut.edu.cn

simple logic circuits. 3. The effect of correlation on circuits is investigated to improve accuracy by decreasing the correlation between intermediate bitstreams.

This paper proceeds as follows. In Section 2, the basic stochastic functions, the stochastic division, and square root circuits are reviewed. Section 3 presents the proposed division and square root circuits. Section 4 reports the experimental results. Section 5 concludes this paper.

2. BACKGROUND

2.1. Stochastic Logic and Correlation

Given two stochastic inputs *A* and *B*, a simple AND gate functions as a stochastic multiplier. Fig. 1(a) shows two input bitstreams *A*=010101101001 and *B*=110101110110 respectively encoding values P(*A*)=6/12, P(*B*)=8/12, and the output bitstream is obtained as *C*=01010110000 encoding the value P(*C*)=4/12, satisfying $C=A\times B$.

The accuracy of SC generally relies on the independence of the input bitstreams. Therefore, most designs avoid correlation, and D flip-flops (DFFs) are the common element to decrease the correlation. However, [15] introduced a measure of correlation among bitstreams and analyzed circuits with correlated inputs. It has been shown that the correlation is not always harmful in SC and it enables logic gates to perform new functions. The bitstream correlation is called a stochastic computing correlation (SCC) and is defined as (1).

$$\mathrm{SCC}\left(S_x,S_y\right)=\begin{cases}\dfrac{\delta\left(S_x,S_y\right)}{\min\left(P\left(S_x\right),P\left(S_y\right)\right)-P\left(S_x\right)P\left(S_y\right)}, & \left(\delta\left(S_x,S_y\right)>0\right)\\ 0, & \left(\delta\left(S_x,S_y\right)=0\right)\\ \dfrac{\delta\left(S_x,S_y\right)}{P\left(S_x\right)P\left(S_y\right)-\max\left(P\left(S_x\right)+P\left(S_y\right)-1,0\right)}, & \left(\delta\left(S_x,S_y\right)<0\right)\end{cases} \tag{1}$$

where $\delta(S_x,S_y)=P(S_x \wedge S_y)-P(S_x)P(S_y)$. The value of SCC is between -1 and 1. When SCC=0, it indicates that the bitstreams are ideally independent. When SCC is 1 or -1, it indicates that the correlation between bitstreams reaches the maximum. By sharing an RNG, bitstreams with maximal correlation can be obtained [16]. Fig. 1(c) shows that the AND gate can realize the function of MIN(*A*,*B*) when its input bitstreams have a maximal correlation. The remaining logic gates in Fig. 1 implement the functions when the input bitstreams of OR and XOR gates are independent and maximally correlated, respectively.

2.2. Stochastic Division Circuits

CORDIV: CORDIV in [9] consists of an SNG unit, a MUX, and a padding unit, as shown in Fig. 2(a). The SNG composed of one RNG and two comparators produces the bitstreams for divisor and dividend. By sharing an RNG between two comparators, the divisor and dividend bitstreams with a maximal correlation are generated. For division, that is quotient=dividend/divisor, the divisor is always greater than the dividend because the input and output values are limited to [0,1]. In the CORDIV design, the dividend is connected to the '1' port of a MUX and the divisor is connected to the select port of the MUX. The result of the division is obtained by using conditional probability.

SSDIV: The stochastic divider in [13], as shown in Fig. 2(b), includes one saturating subtractor and one JKFF. The subtractor includes a NOT gate and an AND gate, which takes the advantage of the correlation among input bitstreams. As shown in [10], the SSDIV is almost the same as the CORDIV in area, delay, power consumption, and accuracy.

2.3. Stochastic Square Root Circuit

BISQRT-S-JK: BISQRT-S-JK in [14] is a low-cost square root circuit, as shown in Fig. 2(c). For values in the range of [0,1], the output is greater than or equal to its input value, which means that the probability of 1s in the input bitstream must be smaller than or equal to the probability of 1s in the output bitstream. A stochastic insertion method was used to design the BISQRT-S-JK to realize the square root function by using a JKFF and a MUX.

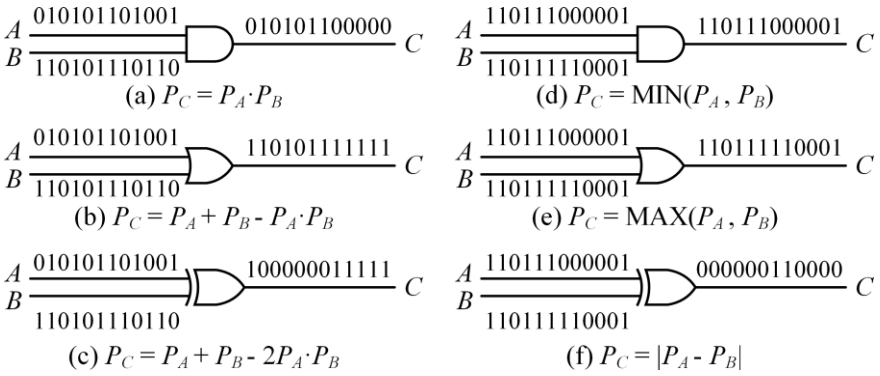

Fig. 1 (a), (b), and (c) Stochastic operators with uncorrelated input bitstreams. (d), (e), and (f) Corresponding operators with correlated input bitstreams.

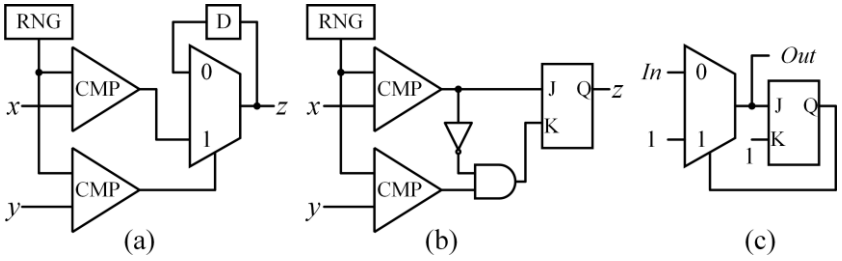

Fig. 2 Existing stochastic division and square root circuits. (a) CORDIV [9]. (b) SSDIV [10]. (c) BISQRT-S-JK [11].

3. PROPOSED DESIGNS

3.1. Stochastic Division Circuit

As shown in Fig. 3, x and y are the two input values of a divider. X and Y are the corresponding stochastic bitstreams generated by sharing an RNG to get the maximal correlation. Thus, the AND and XOR gates can respectively realize the MIN function and absolute subtraction. Specifically, the output of the AND gate gives MIN(X,Y) and is connected to the J port of a JKFF, while the output of the XOR gate gives $|X-Y|$ that is connected to the K port. In SC, a JKFF outputs the value $Q=J/(J+K)$ [13]. Therefore, the output of the proposed divider is z=MIN(X,Y)/(MIN(X,Y)+$|X-Y|$). It specifically contains two cases about X and Y.

Case1: $x \geqslant y$, x is the divisor, y is the dividend, $z=Y/(Y+X-Y)=Y/X$.

Case2: $x<y$, y is the divisor, x is the dividend, $z=X/(X+Y-X)=X/Y$.

It can thus be concluded that the function of the proposed division circuit is MIN(X,Y)/MAX(X,Y).

In current dividers, the divisor and dividend need to be distinguished in advance, and then connected to their corresponding input ports. This is difficult, if not impossible, in larger circuits, especially in an intermediate computation process. The proposed division circuit solves this problem, by using the bitstream correlation. That means no matter which one of the two inputs is the divisor or dividend, the proposed divider can always produce the result of MIN(X,Y)/MAX(X,Y).

The input bitstreams of the JKFF in Fig. 3are respectively denoted as S_J and S_K. Ignore the delay element (DE) for decorrelation, then we have $\delta(S_J,S_K)=P(S_J \wedge S_K)-P(S_J)P(S_K)=-P(S_J)P(S_K)$. According to (1), the SCC between S_J and S_K is computed as (2).

$$\begin{aligned} \mathrm{SCC}(S_J,S_K) &= \frac{\delta(S_J,S_K)}{P(S_J)P(S_K)-\max(P(S_J)+P(S_K)-1,0)} \\ &= \frac{-P(S_J)P(S_K)}{P(S_J)P(S_K)-\max(P(S_J)+P(S_K)-1,0)} \\ &= -1 \end{aligned} \tag{2}$$

This result indicates that the input bitstreams of the JKFF have the maximal negative correlation. To obtain more accurate results for the proposed divider, they are decorrelated by inserting a DE composed of DFFs in the J port.

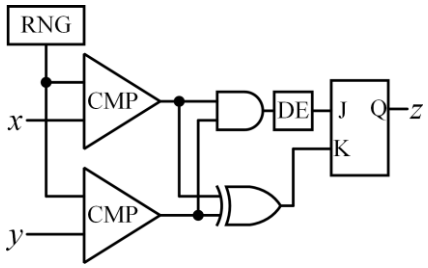

Fig. 3 The proposed stochastic division circuit.

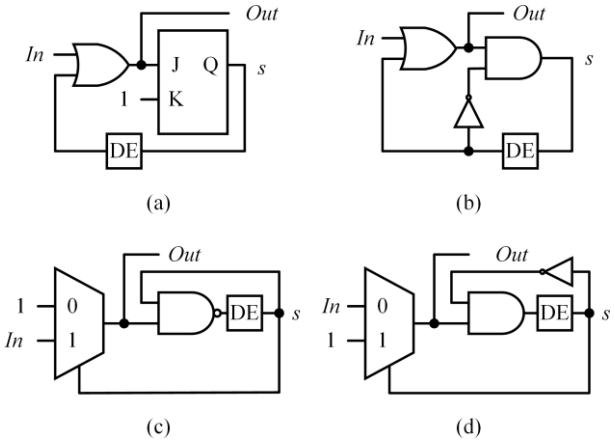

Fig. 4 Four proposed stochastic square root circuits. (a) SSRC-A. (b) SSRC-B. (c) SSRC-C. (d) SSRC-D.

3.2. Stochastic Square Root Circuits

Four stochastic square root circuits (abbreviated as SSRC-A, SSRC-B, SSRC-C, SSRC-D) are proposed by using basic logic components, as shown in Fig. 4, where P_{In} and P_{Out} denote the values of *In* and *Out* ports, and P_s is an intermediate value marked for illustration.

SSRC-A: SSRC-A in Fig. 4(a) consists of three components, an OR gate, a JKFF, and a DE. The *K* port of the JKFF is connected to a full '1' bitstream. The DE is used to improve the accuracy of the circuit by reducing bitstream correlation. The function of SSRC-A is shown as follows.

The output of the OR gate is

$$\begin{aligned} &P_{In} + P_S - P_{In} \cdot P_S = P_{Out} \\ &\Rightarrow P_S = \frac{P_{Out}}{P_{Out}+1} \end{aligned} \tag{3}$$

Therefore, the output of the JKFF is

$$\begin{aligned} &P_{In} + \frac{P_{Out}}{P_{Out}+1} - P_{In} \cdot \frac{P_{Out}}{P_{Out}+1} = P_{Out} \\ &\Rightarrow P_{Out} = \sqrt{P_{In}} \end{aligned} \tag{4}$$

SSRC-B: SSRC-B in Fig. 4(b) consists of four components, an OR gate, an AND gate, a NOT gate, and a DE. One port of the OR gate is connected to the input *In*, and the other port is connected to the output of the AND gate (through the DE). The output of the OR gate is connected to one input port of the AND gate, and the other input of the AND gate is connected to its inverted output. The function of SSRC-B is shown as follows.

The output of the AND gate is

$$\begin{aligned} &P_S = P_{Out} \cdot (1 - P_S) \\ &\Rightarrow P_S = \frac{P_{Out}}{P_{Out}+1} \end{aligned} \tag{5}$$

Therefore, the output of the OR gate is

$$\begin{aligned} & P_{In}+P_S-P_{In}\cdot P_S=P_{Out} \\ & \Rightarrow P_{In}+\frac{P_{Out}}{P_{Out}+1}-P_{In}\cdot\frac{P_{Out}}{P_{Out}+1}=P_{Out} \\ & \Rightarrow P_{Out}=\sqrt{P_{In}} \end{aligned} \tag{6}$$

SSRC-C: SSRC-C in Fig. 4(c) consists of three components, a MUX, a NAND gate and a DE. The '0' port of the MUX is connected to a full '1' bitstream, the '1' port is connected to the input bitstream, and the selected port is connected to the bitstream processed by the NAND gate. The function of SSRC-C is shown as follows.

The output of the NAND gate is

$$\begin{aligned} & P_S=1-P_{Out}\cdot P_S \\ & \Rightarrow P_S=\frac{1}{P_{Out}+1} \end{aligned} \tag{7}$$

Therefore, the output of the MUX is

$$\begin{aligned} & P_S\cdot P_{In}+(1-P_S)\cdot 1=P_{Out} \\ & \Rightarrow \frac{1}{P_{Out}+1}\cdot P_{In}+\left(1-\frac{1}{P_{Out}+1}\right)\cdot 1=P_{Out} \\ & \Rightarrow P_{Out}=\sqrt{P_{In}} \end{aligned} \tag{8}$$

SSRC-D: SSRC-D is similar to SSRC-C, as shown in Fig. 4(d). The difference between SSRC-C and SSRC-D lies in the position of the input In of the MUX and the position of the NOT gate. The function of SSRC-D is shown as follows.

The output of the AND gate is

$$\begin{aligned} & P_S=P_{Out}(1-P_S) \\ & \Rightarrow P_S=\frac{P_{Out}}{P_{Out}+1} \end{aligned} \tag{9}$$

Therefore, the output of the MUX is

$$\begin{aligned} & P_S\cdot 1+(1-P_S)\cdot P_{In}=P_{Out} \\ & \Rightarrow \frac{P_{Out}}{P_{Out}+1}\cdot 1+\left(1-\frac{P_{Out}}{P_{Out}+1}\right)\cdot P_{In}=P_{Out} \\ & \Rightarrow P_{Out}=\sqrt{P_{In}} \end{aligned} \tag{10}$$

The premise of the above analysis process is that these logic units can accurately realize their functions. For the OR, AND, MUX, and JKFF gates, the input bitstreams are expected to be independent, which means that the correlation of the input bitstreams must be reduced as much as possible. Therefore, a DE is added to each circuit to reduce the correlation. In Section 4, the effects of the number of DFFs in each DE on the accuracy of the proposed circuits are explored.

4. EXPERIMENT AND ANALYSIS

As described in prior research, the energy efficiency of SC circuits often falls short of the binary counterparts when the data width is beyond 8 bits [17]. This work focuses on the designs with 8-bit SNGs that produce 255-bit SNs.

4.1. Accuracy Improvement

We studied the effect of the number of DFFs in the DE on the accuracy of the proposed division and square root circuits by considering the mean square error (MSE) and mean absolute error (MAE) as evaluation indicators. Because LFSRs are used in the RNGs in this work, there are fluctuation errors due to seed selection and feedback polynomials. 2000 Monte Carlo experiments are conducted to eliminate the randomness and the average values are taken as the final results.

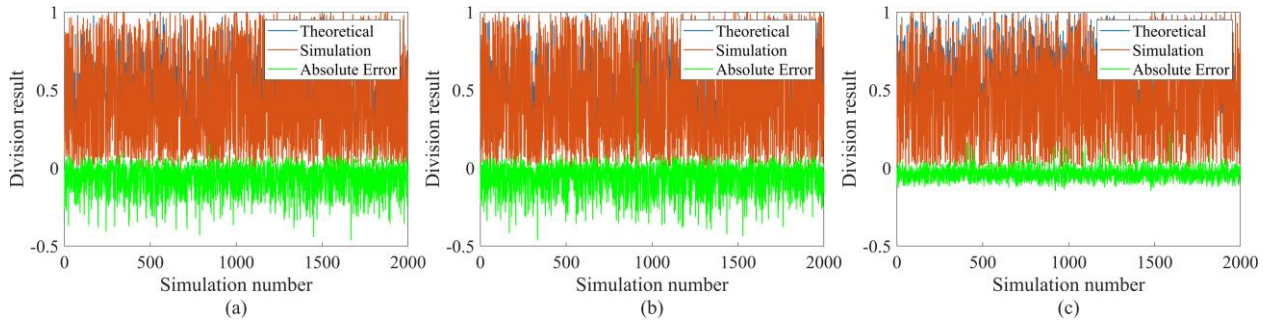

Fig. 5 Division accuracy comparison between (a) CORDIV. (b) SSDIV. (c) The proposed division circuit. The absolute error is obtained as the difference between the simulation result and the theoretical result.

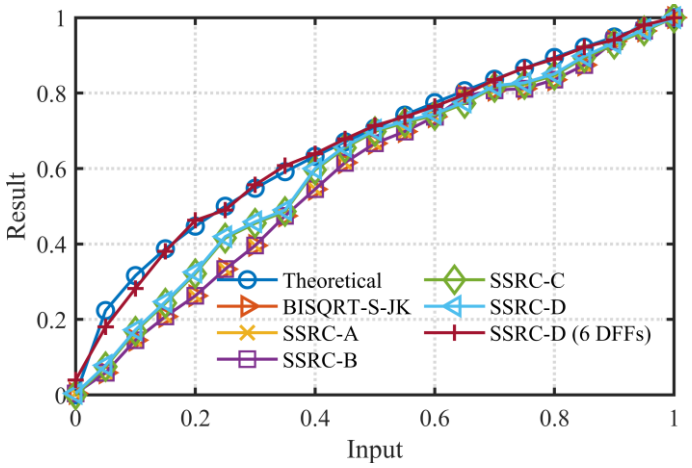

Fig. 6 Accuracy comparison of square root circuits.

For the proposed division circuit, 2000 pairs of x and y are randomly sampled in the range of [0,1] to generate the input bitstreams. TABLE 1 shows the MSE and MAE of the results by inserting different numbers of DFFs in the DE. The experimental results show that the computing accuracy can be improved by appropriately increasing the number of DFFs. However, excessive DFFs will not further improve its accuracy but will incur a larger circuit area. As a trade-off between hardware cost and accuracy, two DFFs are preferred in the DE for the proposed division circuit.

For SSRC-A, the JKFF has an initial value to be sent to the OR gate as an initial bit to work with the first bit of bitstream P_{In} to ensure the correct operation of the circuit. In this case, the number of DFFs in the DE can be 0 for SSRC-A. For the remaining three square root circuits, the bit to work with the first bit of P_{In} is provided by a DFF in the DE. This means that the number of DFFs in the DE is at least one. As listed in TABLE 1, for a 255-bit input stream, the MSE of the proposed square root circuits is lowered as the number of DFFs increases. However, excessive DFFs will not further improve the accuracy, similar to the proposed division circuit.

4.2. Accuracy Comparison

The padding length of CORDIV is 1 in [12], and the number of DFFs is 2 for the proposed division circuit. As shown in Fig. 5, the horizontal axis represents 2000 pairs of x and y, and the vertical axis represents the division result of each pair of x and y. Compared with CORDIV and SSDIV, the proposed stochastic division circuit produces the minimum error peaks and more stable results for the computed quotients; thus, it presents the highest agreement between the simulation and theoretical results. TABLE 2 reveals that the MSE of the proposed division circuit with 2 DFFs is lowered by 76% compared with CORDIV and SSDIV, with a slightly improved hardware cost. These results show that the proposed division circuit is more accurate than CORDIV and SSDIV.

The data in TABLE 2 show that SSRC-A and SSRC-B have the same accuracy as BISQRT-S-JK. The MSEs of SSRC-C and SSRC-D are lowered by 45% and 48%, respectively, compared to that of BISQRT-S-JK. Fig. 6 shows the accuracy comparison of different stochastic square root circuits. The horizontal axis represents the input values that are sampled from 0

to 1 with a step

Table 1 Accuracy of proposed stochastic circuits with different numbers of DFFs ($\times 10^{-2}$)

The number of DFFs in each DE			0	1	2	3	4	5	6	7
Proposed division circuit		MSE	1.26	1.44	0.31	0.29	0.29	0.32	0.33	0.35
		MAE	7.75	9.88	4.56	4.10	4.01	4.13	4.28	4.39
Proposed square root circuits	SSRC-A	MSE	1.09	0.73	0.55	0.46	0.42	0.39	0.40	0.40
		MAE	8.40	6.94	5.84	5.45	5.03	5.00	4.99	4.99
	SSRC-B	MSE	-	1.09	0.73	0.55	0.46	0.42	0.39	0.39
		MAE	-	8.40	6.95	5.84	5.44	5.03	5.00	5.00
	SSRC-C	MSE	-	0.60	0.32	0.18	0.18	0.14	0.10	0.13
		MAE	-	5.95	4.53	2.89	2.89	2.81	2.13	2.68
	SSRC-D	MSE	-	0.57	0.29	0.14	0.14	0.10	0.06	0.07
		MAE	-	5.80	4.25	2.61	3.00	2.27	1.74	1.92

Table 2 Hardware and accuracy comparison of the proposed stochastic division and square root circuits with previous work

Design		Area (um^2)			Power (uW)	Delay (ns)	MSE ($\times 10^{-2}$)	MAE ($\times 10^{-2}$)
		SNG	Kernel	Total				
Division	CORDIV	67.38	3.36	70.74	5.11	0.77	1.28	7.90
	SSDIV	67.38	1.95	69.33	4.49	0.69	1.28	7.83
	Proposed	67.38	12.53	79.91	5.28	0.90	0.31	4.56
Square root	BISQRT-S-JK	53.62	4.94	58.56	3.99	0.87	1.09	8.40
	SSRC-A	53.62	4.94	58.56	3.99	0.87	1.09	8.40
	SSRC-B	53.62	3.71	57.33	3.94	0.85	1.09	8.40
	SSRC-C	53.62	5.65	59.27	4.10	0.83	0.60	5.95
	SSRC-D	53.62	5.65	59.27	4.10	0.83	0.57	5.80

of 1/20. That is, there are 21 inputs within the interval of [0,1]. The vertical axis represents the output results of the square root circuits. It can be seen that the results of SSRC-D with 6 DFFs in the DE almost coincide with the theoretical results.

4.3. Hardware Cost

All the proposed circuits are synthesized with TSMC's 40nm library at 100MHz by the Synopsys Design Compiler. The power, area, and delay (critical path delay) are listed in TABLE 2. The proposed division circuit is synthesized with 2 DFFs, the padding length of CORDIV is 1, and the number of DFF in the DE of the proposed SSRC is the lowest. It shows that the SNG takes up most area of an SC circuit. The area of the proposed divider is larger than those of CORDIV and SSDIV because 2 DFFs in the DE are used for decreasing the correlation. The kernel area of the proposed division circuit is considered to contain a JKFF and DE unit, while SSDIV (CORDIV) considers JKFF (MUX) only. The proposed SSRC-A incurs the same hardware cost as BISQRT-S-JK. Although there is no performance improvement, this design provides a new implementation method and can achieve the same performance, providing more options for stochastic square root operation. For the proposed SSRC-B, under the same accuracy, its area is 57.33 um^2, which is a slight improvement in area compared with 58.56 um^2 of BISQRT-S-JK. This is the same for power and delay.

4.4. Applications

To verify the performance of the proposed circuits in actual applications, they are applied to image processing algorithms. Considering the feature of division and square root circuits, contrast stretching and gamma correction algorithms are adopted respectively.

Contrast stretching: As a method of image enhancement, contrast stretching improves the image contrast by changing the range of gray values of image pixels, which belongs to the gray transformation operation. Different from the more complex

histogram equalization, it lowers the difficulty in enhancing the image by applying a linear scaling function to pixel values,

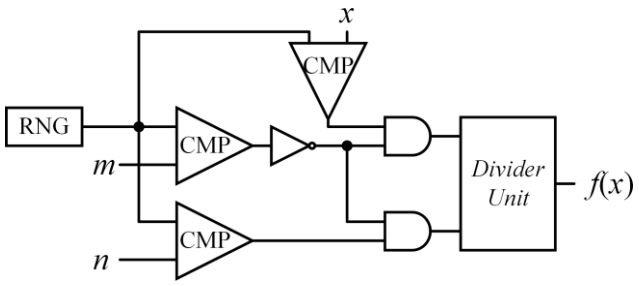

Fig. 7 Architecture of contrast stretching function [10].

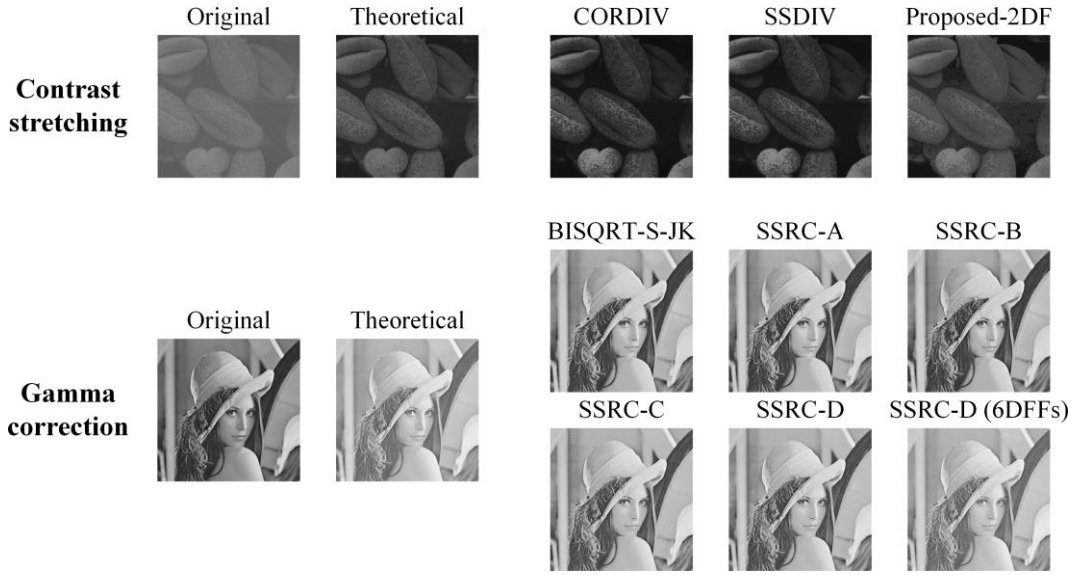

Fig. 8 Image processing results of different implementations.

and the function is shown as (11).

$$f(x)=\begin{cases}0, & 0\le x<m\\ \dfrac{x-m}{n-m}, & m\le x\le n\\ 1, & x>n\end{cases} \tag{11}$$

where 0<m<n<1. The function is presented in Fig. 7 [13]. *Divider Unite* is implemented with the proposed divider, CORDIV, and SSDIV respectively. To evaluate the performance of the contrast stretching circuits, *m* is set to be 0.3, and *n* is set to be 0.8.

Gamma correction: The actual output image will deviate in brightness due to the display or other reasons in a system. Gamma correction is a method to code and decode the brightness in video and still image systems to correct the deviation of the image and make the image look more in line with the characteristics of human eyes [9]. It is a nonlinear operation and is defined by a power-law expression as (12).

$$Out = In^{\gamma} \tag{12}$$

We apply a value of $\gamma = 0.5$, and the gamma correction function can be written as (13).

$$f(x) = x^{0.5} = \sqrt{x} \tag{13}$$

Hence, the function can be realized with the square root circuit directly.

For the two applications above, bitstreams with a length of 255 bits are exploited to process images. The application results are shown in Fig. 8. The Peak Signal to Noise Ratio (PSNR) and MSE are computed to compare the accuracy of the circuits. The average PSNR and MSE are computed with 1000 trials. The circuits are also synthesized to compare their hardware cost. The experimental data of contrast stretching are shown in Table 3. Compared with the CORDIV (SSDIV)

method, the proposed design has a 91% (90%) lower MSE and 47% (45%) improvement in PSNR, with a slight area increase, which is an acceptable trade-off. The accuracy comparison of gamma correction is shown in Table 4. It indicates that the proposed SSRC-A and SSRC-B have the same accuracy as BISQRT-S-JK, while SSRC-C and SSRC-D are lowered 50% and 52% in MSE, respectively. Furthermore, the MSE of the SSRC-D with 6 DFFs can be lowered by 97% compared with that of BISQRT-S-JK. As the application of gamma correction can be directly implemented with the square root circuit, and no more extra elements. Thus, the hardware cost of gamma correction is the same as the data in Table 2.

Table 3 Performance comparison of contrast stretching

Design	MSE ($\times 10^{-3}$)	PSNR	Area (um^2)	Power (uW)	Delay (ns)
Proposed	0.57	32.42	99.66	5.79	1.00
CORDIV	6.21	22.07	90.20	5.51	1.05
SSDIV	5.76	22.40	91.46	5.01	0.77

Table 4 Performance comparison of gamma correction

Design	MSE ($\times 10^{-3}$)	PSNR
SSRC-A	9.92	20.03
SSRC-B	9.92	20.03
SSRC-C	4.71	23.26
SSRC-D	4.51	23.46
SSRC-D (6DFF)	0.28	35.47
BISQRT-S-JK	9.92	20.03

5. CONCLUSION

A stochastic divider is proposed by using the correlation among input bitstreams to eliminate the need for distinguishing the divisor and dividend. Four stochastic square root circuits are also designed for a higher accuracy by inserting a DE to reduce the intermediate bitstream correlation. Experimental results show that the exploitation of the bitstream decorrelation can simplify the circuits in SC with improved accuracy.

ACKNOWLEDGEMENT

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.

REFERENCES

[1] B. R. Gaines, "Stochastic computing," in *Proceedings of the Fall Joint Computer Conference*, Atlantic City, New Jersey, 1967, pp. 149-156, 1465505: ACM.

[2] W. J. Poppelbaum, C. Afuso, and J. W. Esch, "Stochastic computing elements and systems," in *Proceedings of the Fall Joint Computer Conference*, Anaheim, California, 1967, pp. 635-644, 1465696: ACM.

[3] A. Alaghi, W. Qian, and J. P. Hayes, "The Promise and Challenge of Stochastic Computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 37, no. 8, pp. 1515-1531, 2018.

[4] Á. Rák, B. G. Soós, and G. Cserey, "Stochastic bitstream-based CNN and its implementation on FPGA," *International Journal of Circuit Theory and Applications,* vol. 37, no. 4, pp. 587-612, 2009.

[5] C. Subhasri, B. R. Jammu, L. Guna Sekhar Sai Harsha, N. Bodasingi, and V. R. Samoju, "Hardware ‐ efficient approximate logarithmic division with improved accuracy," *International Journal of Circuit Theory and Applications,* vol.

49, no. 1, pp. 128-141, 2020.

[6] R. Omidi and S. Sharifzadeh, "Design of low power approximate floating‐point adders," *International Journal of Circuit Theory and Applications,* vol. 49, no. 1, pp. 185-195, 2020.

[7] 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, 2021.

[8] B. D. Brown and H. C. Card, "Stochastic neural computation I: Computational elements," (in English), *IEEE Transactions on Computers,* Article vol. 50, no. 9, pp. 891-905, Sep 2001.

[9] W. K. Qian, X. Li, M. D. Riedel, K. Bazargan, and D. J. Lilja, "An Architecture for Fault-Tolerant Computation with Stochastic Logic," *IEEE Transactions on Computers,* vol. 60, no. 1, pp. 93-105, Jan 2011.

[10] K. K. Parhi and Y. Liu, "Computing Arithmetic Functions Using Stochastic Logic by Series Expansion," *IEEE Transactions on Emerging Topics in Computing,* vol. 7, no. 1, pp. 44-59, 2019.

[11] B. R. Gaines, "Stochastic computing systems," in *Advances in Information Systems Science*, J. T. Tou, Ed. (Advances in Information Systems Science: Springer, Boston, MA, 1969, pp. 37-172.

[12] T. Chen and J. P. Hayes, "Design of Division Circuits for Stochastic Computing," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2016, pp. 116-121.

[13] S.-I. Chu, "New Divider Design for Stochastic Computing," *IEEE Transactions on Circuits and Systems II: Express Briefs,* pp. 1-5, Mar 2020.

[14] D. Wu, R. Yin, and J. S. Miguel, "In-Stream Correlation-Based Division and Bit-Inserting Square Root in Stochastic Computing," *IEEE Design & Test,* pp. 1-1, 2021.

[15] A. Alaghi and J. P. Hayes, "Exploiting correlation in stochastic circuit design," in *2013 IEEE 31st International Conference on Computer Design (ICCD)*, 2013, pp. 39-46.

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

[17] V. T. Lee, A. Alaghi, R. Pamula, V. S. Sathe, L. Ceze, and M. Oskin, "Architecture Considerations for Stochastic Computing Accelerators," (in English), *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 37, no. 11, pp. 2277-2289, Nov 2018.