Accelerating Stochastic Computing Using Deterministic Halton Sequences

Zhendong Lin, Guangjun Xie, Wenbing Xu, Jie Han, *Senior Member, IEEE*, and Yongqiang Zhang

***Abstract*—Deterministic approaches have recently been developed for accurate computation in stochastic computing (SC). They, however, suffer a long operation time. Fortunately, for applications that do not require completely accurate processing results, such as image processing, the time can significantly be reduced due to the better progressive precision in the bit-streams generated by these approaches. That means a computation can be terminated in time when its output accuracy is acceptable. Due to the fast convergence property of low-discrepancy sequences, we propose three deterministic Halton sequence (DHS)-based stochastic number generators (SNGs) for the first time by using, respectively, prime length, rotation, and clock division for accelerating computation in SC. Experimental results show that the proposed designs are more efficient than their counterparts. For multiplication, the proposed DHS-based designs perform up to 32× faster than prior works for a mean absolute error of 0.1%. The speedup reaches 128× for an edge detection algorithm. Three stochastic circuits are then designed by using the proposed DHS-based SNGs for the Bernsen binarization algorithm, which lead to more accurate results than existing designs at the same bit-stream length. Finally, the proposed designs show an excellent fault-tolerance against bit flipping errors.**

***Index Terms*—Deterministic approach, Halton sequence, Stochastic computing, Progressive precision, Stochastic number generator, Fault-tolerance.**

I. Introduction

STOCHASTIC computing (SC) has emerged as an alternative to conventional weighted binary computing for its low-cost computing core, low power consumption, and inherent fault-tolerance. Recently, SC has been successfully applied in various fields, such as image processing [1], deep neural networks (DNNs) [2], and non-linear functions [3]. A common feature of these applications is that their output results do not need to be completely accurate or, in other words, a slight inaccuracy is acceptable.

Typical computation in SC is performed on unary bit-streams encoding stochastic numbers (SNs) and produced by stochastic number generators (SNGs). Considering the proportion of 1s, for instance, bit-streams X=11101000 and Y=1010 denote P_X=P_Y=1/2 because they both contain 50% of the bits as 1s. Neither their length nor the positions of 1s and 0s are fixed [4]. Generally, the longer the bit-stream is, the higher the accuracy. However, a longer bit-stream leads to the drawback of a longer processing time.

Recently, deterministic approaches (prime length, rotation, and clock division) have been developed for performing completely accurate computation [5]. However, the latency is much higher than classical SC approaches. For example, when two n-bit streams are used for multiplication, the output bit-stream length (BSL) has to be 2^{2n} if an accurate result is required [6], which makes the long latency a key issue.

The low-discrepancy Halton sequence is widely used in quasi-Monte Carlo simulation and fast convergence is one of its main advantages [7]. In SC, this advantage enables Halton sequence-based SNGs to produce bit-streams with a shorter BSL and higher accuracy, which is defined as a progressive precision property. The main contributions of this brief are summarized as follows: 1) Three deterministic Halton sequence (DHS)-based SNGs are proposed for the first time by using deterministic approaches. 2) In applications for which a slight inaccuracy is acceptable, the processing time could be significantly reduced by terminating calculations early, and thus reducing the latency and energy consumption. Moreover, the hardware overhead of the proposed design is lower than that of previous designs. 3) The proposed designs are then applied to two image processing algorithms to illustrate their performance; experimental results show that they outperform prior designs in terms of processing time, accuracy, and fault-tolerance.

This brief proceeds as follows. Section II presents the basic concepts of Halton sequence and deterministic approaches. We propose DHS-based SNGs and demonstrate their performance in Section III. Section IV presents the experimental results for two image processing cases. Section V concludes this brief.

II. Background

A. Low-Discrepancy Halton Sequences

In SC, a smaller absolute value of stochastic computing correlation (SCC) usually means a higher accuracy in the computed result [8]. Low-discrepancy sequences-based bit-

This work was supported by the Fundamental Research Funds for the Central Universities of China under Grant No. JZ2020HGTA0085 and No. JZ2020HGQA0162.

(*Corresponding author: Yongqiang Zhang*)

Z. Lin, G. Xie, W. Xu and Y. Zhang are with the School of Electronic Science and Applied Physics, Hefei University of Technology, Hefei 230009, China (e-mail: zdlin@mail.hfut.edu.cn; gjxie8005@hfut.edu.cn; wbxu@mail.hfut.edu.cn; ahzhangyq@hfut.edu.cn)

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

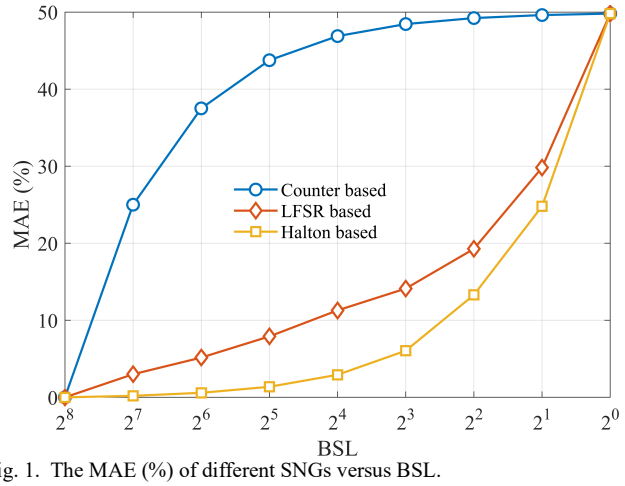

Fig. 1. The MAE (%) of different SNGs versus BSL.

streams representing given target values are SNs that are better equidistributed than pseudo-random numbers with a lower SCC and faster convergence [9]. Halton and Sobol sequences are two commonly used low-discrepancy sequences [10]. Due to their inherent better progressive precision property, the computation accuracy is increasing with processing time. We consider the use of Halton sequences in stochastic circuits since the hardware overhead for generating Sobol sequences is higher than that for Halton sequences [11].

The authors in [7] proposed a low-discrepancy Halton sequence-based SNG. To acquire less correlated bit-streams, different binary-coded base-*b* counters are required for different SNGs as a number source (*b* is a prime number), which leads to a substantial hardware overhead. The 1s and 0s in the bit-stream generated by the Halton sequence-based SNG are evenly arranged, which makes it converge faster to its target value than others. To illustrate the low-discrepancy property of Halton sequences, we evaluate the progressive precision property of different bit-streams generated by the counter, LFSR, and base-2 Halton sequence-based SNGs respectively, as shown in Fig. 1. We use 8-bit SNGs and intercept half of the BSL each time. The mean absolute error (MAE) is evaluated and the SN ranges from 0-255/256. As can be seen in this figure, the Halton sequence-based SNG produces the bit-streams with the lower MAE that the counter and LFSR based SNGs cannot achieve.

B. Deterministic Approaches

Three deterministic approaches (using prime length, rotation, and clock division) were developed in [5] for accurate computing. The SCC between two bit-streams becomes 0 because each bit in one bit-stream interacts with every bit in the other bit-stream for these approaches, similar to the convolution operation in mathematics. It is noteworthy that the SCC between their subsequences is much closer to 0 than that of classical pseudo-random bit-streams, so it leads to more accurate results. Several designs were developed in [6] to reduce the processing time. The long latency, however, is still a serious drawback in these designs, which results in a high energy consumption value. In [12] and [13], a parallel SNG and a resolution splitting design are used for accelerating SC, respectively. Both of them suggest the use of deterministic approaches, but they suffer from high hardware overhead that offset the advantage of SC.

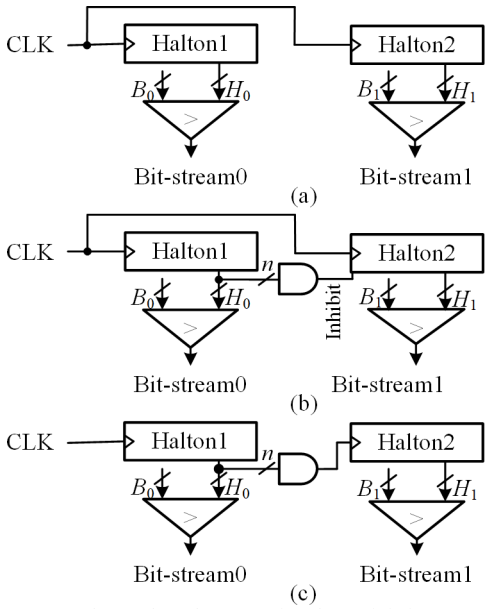

Fig. 2. The proposed DHS-based SNGs using deterministic approaches based on (a) Prime length. (b) Rotation. (c) Clock division.

III. The Proposed DHS-Based SNGs

A. The Proposed Architecture

We suggest only use base-2 Halton sequences for all SNGs since the Halton sequence generator will be reduced to a simple binary counter so that its hardware overhead could be significantly reduced. The correlation between bit-streams will be eliminated by using deterministic approaches. Fig. 2 shows three architectures for the proposed DHS-based SNGs by using the three deterministic approaches described in Section II. The number source of Halton1 keeps counting all the time in all three designs. For the prime length approach, we control the count status of counter2 (the number source of Halton2 in Fig. 2(a)) by one bit less to implement the relative prime bit length. For instance, assume the bit-streams are 8-bit long, counter2 is restarted with the state of 00000000 if the current state is 11111110. Then the two bit-streams repeat all the time, thus the complete BSL is $2^8\times(2^8-1)=65280$. As for the rotation and clock division approaches, these counters all range from 00000000 ~ 11111111. When counter1 outputs all 1s, counter2 is inhibited in the rotation approach or counts once in the clock division approach. Thus, the complete BSL in the rotation and clock division approaches is $2^8\times2^8=2^{16}$.

B. SNG Cost Comparison

We compare the hardware overhead of the proposed three DHS-based SNGs and prior designs by using Synopsys Design Compiler, with the 45-nm NanGate library [14]. Synthesized results are shown in TABLE I. All SNGs are of an 8-bit precision. Columns 3-8 show the area, power, delay, area power product (ADP), power delay product (PDP), and energy delay product (EDP) of the three designs with various deterministic approaches. It is evident that, for the three deterministic

TABLE I
THE HARDWARE AREA, POWER, DELAY, ADP, PDP, AND EDP FOR SNGS

Approach	SNG	Area (μm^2)	Power (μW)	Delay (ns)	ADP ($\mu m^2 \times ns$)	PDP ($pJ \times 10^{-3}$)	EDP ($10^{-24} Js$)
Prime length	Counter [5]	156.67	1.89	2.26	354.08	4.26	9.63
	LFSR [6]	193.65	2.36	2.43	470.56	5.73	13.92
	This Work	156.14	1.90	2.24	349.76	4.25	9.52
Rotation	Counter [5]	151.89	1.89	2.54	385.79	4.78	12.13
	LFSR [6]	192.58	2.35	2.43	467.98	5.71	13.88
	This Work	154.28	1.91	2.24	345.59	4.29	9.61
Clock division	Counter [5]	154.01	1.88	2.54	391.20	4.77	12.10
	LFSR [6]	192.58	2.35	2.43	467.98	5.71	13.88
	This Work	153.75	1.90	2.24	344.40	4.25	9.52

TABLE II
THE MAE (%) FOR MULTIPLICATION USING DETERMINISTIC APPROACHES VERSUS BSL

Approach	SNG	2^{16}	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}
Prime length	Counter [5]	**0**	3.1500	4.9200	6.2500	7.1900	7.7800	8.1200	8.3000	8.4000
	LFSR [6]	**0.1000**	0.1700	0.2200	0.2800	0.3300	0.3900	0.4300	0.4800	0.5500
	This work	0	0.0004	0.0019	0.0074	0.0297	**0.1200**	0.4800	1.9900	8.4000
Rotation	Counter [5]	**0**	3.1300	4.8800	6.2000	7.1400	7.7200	8.0600	8.2400	8.3330
	LFSR [6]	0	**0.1100**	0.1700	0.2300	0.2800	0.3400	0.3900	0.4500	0.5100
	This work	0	0.0004	0.0018	0.0073	0.0294	**0.1200**	0.4800	1.9800	8.3300
Clock division	Counter [5]	**0**	12.4500	18.6800	21.7900	23.3500	24.1200	24.5100	24.7100	24.8100
	LFSR [6]	**0**	1.5100	2.4500	3.6600	5.3600	7.2400	9.8900	14.4100	24.8100
	This work	0	**0.0973**	0.2900	0.6800	1.4600	3.0200	6.1300	12.3500	24.8100

TABLE III
THE MAE (%) FOR EDGE DETECTION VERSUS BSL

Approach	SNG	2^{16}	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}
Prime length	Counter [5]	**0.0969**	1.0600	1.5700	1.7800	1.8500	1.8800	1.8900	1.9000	1.9000
	LFSR [6]	0.0966	**0.1000**	0.1100	0.1300	0.1500	0.1800	0.2300	0.3000	0.4000
	This work	0.0969	0.0969	0.0969	0.0970	0.0973	0.0980	0.1000	**0.1000**	1.9000
Rotation	Counter [5]	**0.0957**	1.0600	1.6400	1.8200	1.8700	1.8800	1.8900	1.8900	1.8900
	LFSR [6]	0.0957	**0.1000**	0.1100	0.1300	0.1500	0.1800	0.2300	0.3000	0.4000
	This work	0.0957	0.0957	0.0957	0.0957	0.0957	0.0957	0.0957	**0.0957**	1.8900
Clock division	Counter [5]	**0.0957**	2.2100	2.2100	2.2100	2.2100	2.2100	2.2100	2.2100	2.2100
	LFSR [6]	**0.0957**	0.1700	0.2500	0.3200	0.4600	0.6300	0.8700	1.1800	2.2100
	This work	0.0957	0.0957	0.0957	0.0957	0.0957	0.0957	0.0957	**0.0957**	2.2100
Classical	10-bit LFSR	——	——	——	——	——	——	0.1300	0.5300	0.8800

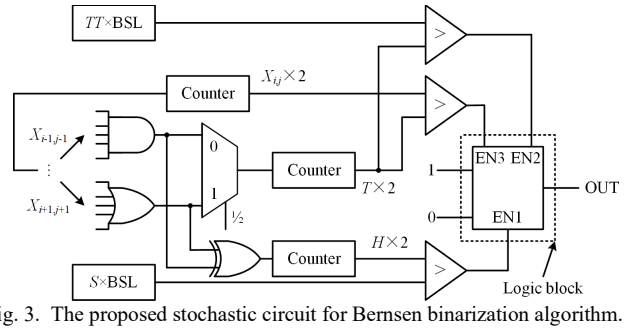

Fig. 3. The proposed stochastic circuit for Bernsen binarization algorithm.


Original Binary
Fig. 4. Original photo and one processed by conventional binary method.

approaches, the hardware cost of the proposed DHS-based designs is the lowest when the ADP, PDP, and EDP are considered. The counter-based designs are a little bit higher than that of the proposed designs, and the LFSR-based designs incur the highest hardware overhead. Thus, the proposed designs have superior performance in terms of hardware overhead.

C. MAE

A half-length bit-stream is used each time to verify that the bit-streams generated by the proposed SNGs have better truncation error properties. We compare the MAE of multiplying two 8-bit precision bit-streams generated by SNGs with various deterministic approaches. Actually, the BSL in the prime length approach is $2^8 \times (2^8-1)$, $2^7 \times (2^8-1)$,$\cdots 2^0 \times (2^8-1)$ by each truncation. For simplicity, we record them as $2^8 \times 2^8 = 2^{16}$, $2^7 \times 2^8 = 2^{15}$,$\cdots$, $2^0 \times 2^8 = 2^8$, as illustrated in TABLE II.

The MAE becomes 0 when the BSL is 2^{16} except for the LFSR-based SNG with the prime length approach. It is more efficient to use the rotation approach, especially for the proposed DHS-based SNG in this work. As BSL decreases, the MAE of our designs grows more slowly when the BSL is greater than 2^{10}. For example, in the prime length approach, the BSL of our design decreases to 2^{11} while it is 2^{16} for the prior designs if a 0.1% error is considered. For the rotation approach, it is 2^{11} in our design while it is respectively 2^{16} and 2^{15} for the counter and LFSR-based SNGs. For the clock division approach, our design also realizes acceptable results by using shorter BSL. Thus, the proposed DHS-based SNGs can dramatically reduce the processing time and further save energy.

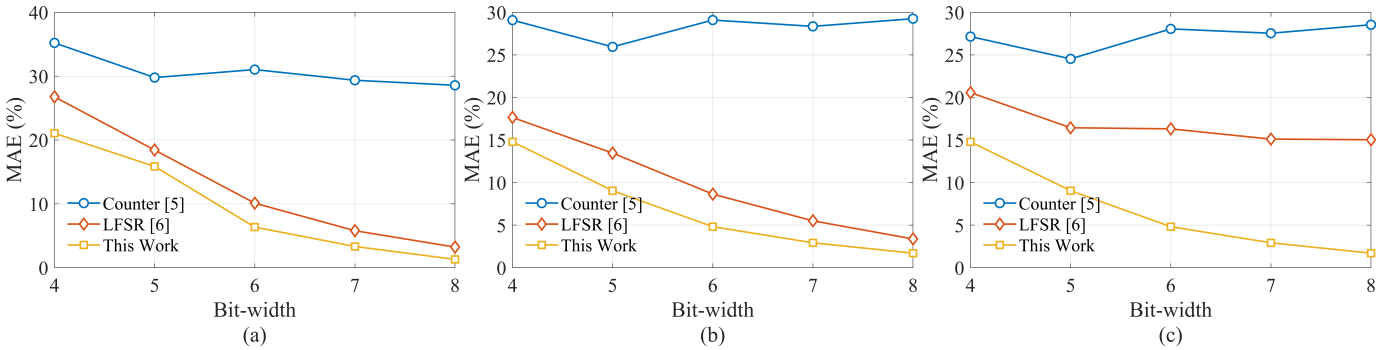

Fig. 5. The MAE (%) for the Bernsen binarization algorithm using (a) Prime length. (b) Rotation. (c) Clock division.

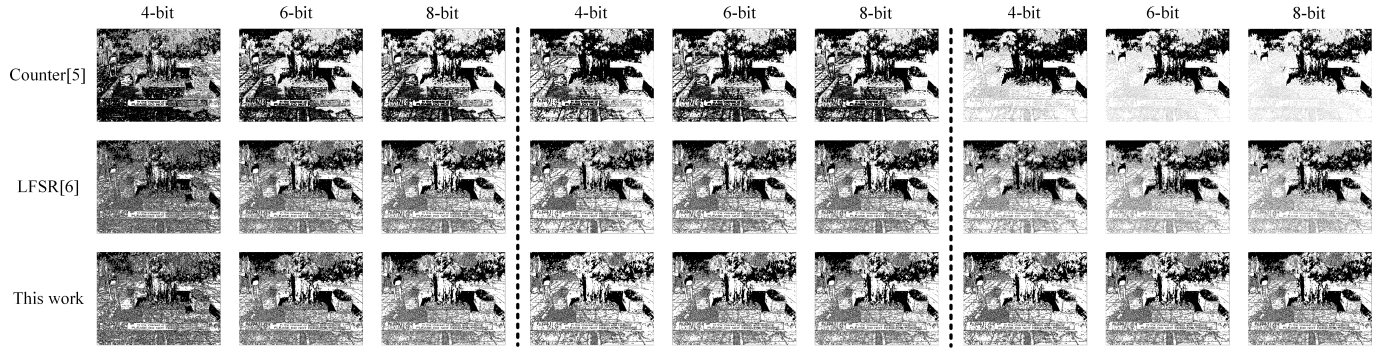

Fig. 6. Part of the processed photos using (a) Prime length. (b) Rotation. (c) Clock division.

IV. Case Studies

We present experimental results of two typical image processing algorithms in this section to highlight the advantages of our designs.

A. Robert's Cross Edge Detection

Robert's cross edge detector [15] is a well-known digital image processing algorithm, which implements

$$Z_{i,j} = 0.5(|X_{i,j} - X_{i+1,j+1}| + |X_{i+1,j} - X_{i,j+1}|), \quad (1)$$

where $X_{i,j} \sim X_{i+1,j+1}$ are the current pixels in the input photo to be processed, $Z_{i,j}$ is the new pixel output generated by this algorithm.

A photo of 128×128 pixels is used to evaluate the performance of various designs. We compare the MAE as listed in TABLE III. All the deterministic SNGs are of an 8-bit precision. If 0.1% error is considered, for the prime length and rotation approaches, the BSL is 2^9 for the proposed DHS-based design, which is only 1/64 and 1/128 of the BSL in the LFSR-based and counter-based design, respectively. For the clock division approach, this ratio is 1/128. Compared with the classical 10-bit LFSR-based design without using deterministic approaches, our designs are also more accurate for the same BSL. Hence, the proposed DHS-based SNGs in this work are much more accurate and efficient.

It turns out that the accuracy of the proposed DHS-based SNGs will not decrease when the BSL is not smaller than 2^9 except the one using the prime length approach. Hence, we conclude that for a MUX (or scaled addition), the output of two n-bit input MUX using the DHS-based SNG is completely accurate when the BSL is not smaller than 2^{n+1}. Because the SNs generated by the Halton-based SNG are uniformly spaced, the input bit-streams of the MUX will be "completely selected" with deterministic approaches if the BSL is not smaller than 2^{n+1}, thus a scaled addition is correctly executed.

B. Bernsen Binarization Algorithm

We present another digital image processing algorithm called the Bernsen binarization algorithm (proposed in [1]) in SC, which is used to mitigate the problem of uneven lighting in photos.

The proposed hardware design is shown in Fig. 3. We set the BSL to 2^{n+1} for accelerating computation as the MUX gate is a key component in this circuit. TT and S are the user defined thresholds, $X_{i-1,j-1} \sim X_{i+1,j+1}$ are the pixel values in a $k \times k$ window centered on $X_{i,j}$ (for the trade-off between accuracy and processing time, we set k=3 for this work). Notice that the AND gate and OR gate obtain maximum and minimum respectively if their inputs are correlated by simply sharing a random number source. The three parameters T, H, and $X_{i,j}$ are doubled because the BSL is 2^{n+1}. The counters are used as stochastic-to-binary converters and connected in the comparators. Then the output will be 0 or 1 via a logical processing structure (that contains three enable signals in this figure) to implement

$$OUT = (EN1 \cdot EN3) + (\overline{EN1} \cdot EN2). \quad (2)$$

A photo with uneven lighting is used as the input for this circuit. Fig. 4 shows the original unevenly lit photo and the image processed by the conventional binary method. Fig. 5 shows the MAE (%) of three SNGs with different deterministic approaches versus the bit-width.

As can be seen from Fig. 5, the larger the bit-width, the more accurate the results are for all designs except for the counter-

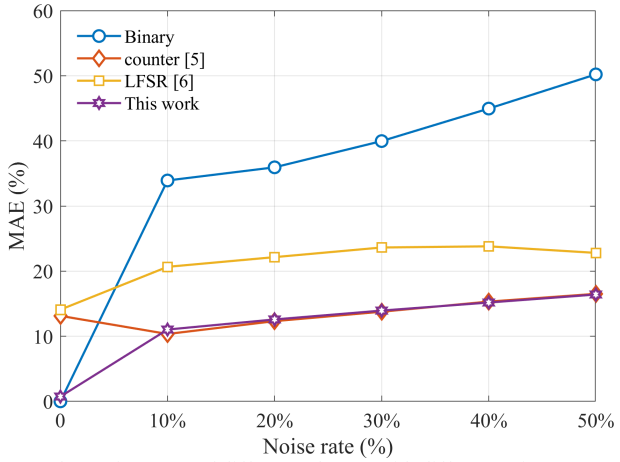

Fig. 7. The performance of different schemes with different noise rate.

based ones. It can be seen that the proposed design works very well for this algorithm. For example, when the bit-width is 8, the error is around 1% by the proposed design for various deterministic approaches. For the prime length approach, our design improves nearly 22× and 2× compared to the counter and LFSR-based designs, respectively, in terms of accuracy. It reaches 17× and 2× accuracy in the rotation approach compared to those two approaches. For the clock division approach, our work achieves nearly 17× and 9× improvement. We can also draw this conclusion from Fig. 6. The details in these photos processed by our designs are clearer in the texture. Therefore, the proposed designs are more effective for this application.

We also evaluate the fault-tolerance by injecting different levels of noise between 0~50% into the computation. The counter, LFSR, and DHS-based SNGs using the rotation approach, and conventional binary method are evaluated. The BSL is 2^{8+1} in all SC implementations. As shown in Fig. 7, the proposed design achieves a lower MAE, so it is more robust than other methods.

V. Conclusion

A circuit in stochastic computing (SC) can generate completely accurate outputs by using deterministic approaches. These methods, however, suffer a long processing time. In applications that a certain inaccuracy is acceptable such as image processing, the latency can be significantly reduced through deterministic approaches because of their inherently superior truncation error property. Due to their fast converging properties, three deterministic low-discrepancy Halton sequence (DHS)-based stochastic number generators (SNGs) are proposed for the first time by respectively using prime length, rotation, and clock division approaches for accelerating the computation in SC. Experimental results show that compared with previous designs, our design can reduce the processing time, provide more accurate, more energy-efficient, and more robust results. Nevertheless, the proposed designs still require a long processing time compared with their binary counterparts. This issue will be addressed in the future work using parallel datapaths in applications such as deep neural networks (DNNs) and other non-linear functions.

References

[1] J. Bersen, "Dynamic thresholding of grey-level images," in Eighth International Conference on Pattern Recognition, Paris, France, 1986, pp. 1251-5.

[2] A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. J. Gross, "Vlsi implementation of deep neural network using integral stochastic computing," *IEEE Trans. Very Large Scale Integr. VLSI Syst.,* vol. 25, no. 10, pp. 2688-2699, Oct, 2017.

[3] A. Ardakani, A. Ardakani, and W. J. Gross, "A regression-based method to synthesize complex arithmetic computations on stochastic streams," in 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 2020, pp. 1-5.

[4] S. Shenoi, "A comparative study on methods for stochastic number generation," College of Engineering and Applied Sciences, University of Cincinnati, Ann Arbor, 2017.

[5] D. Jenson, and M. Riedel, "A deterministic approach to stochastic computation," in 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Austin, TX, USA, 2016, pp. 1-8.

[6] H. Najafi, and D. Lilja, "High quality down-sampling for deterministic approaches to stochastic computing," *IEEE Trans. Emerging Top. Comput.*, pp. 1-1, 2018.

[7] A. Alaghi, and J. Hayes, "Fast and accurate computation using stochastic circuits," in 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 2014, pp. 1-4.

[8] S. A. 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.

[9] I. Dalal, D. Stefan, and J. Harwayne-Gidansky, "Low discrepancy sequences for monte carlo simulations on reconfigurable platforms," in 2008 International Conference on Application-Specific Systems, Architectures and Processors, Leuven, Belgium, 2008, pp. 108-113.

[10] 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.

[11] S. Liu, and J. Han, "Energy efficient stochastic computing with Sobol sequences," in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, Lausanne, 2017, pp. 650-653.

[12] Y. Zhang, R. Wang, X. Zhang, Y. Wang, and R. Huang, "Parallel hybrid stochastic-binary-based neural network accelerators," *IEEE Transactions on Circuits and Systems II-Express Briefs*, pp. 1-1, 2020.

[13] M. H. Najafi, S. R. Faraji, B. Z. Li, D. J. Lilja, and K. Bazargan, "Accelerating deterministic bit-stream computing with resolution splitting," in 20th International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, 2019, pp. 157-162.

[14] "Nangate open cell library," https://projects.si2.org.

[15] A. Alaghi, L. Cheng, and J. Hayes, "Stochastic circuits for real-time image-processing applications," in 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), Austin, TX, USA, 2013, pp. 1-6.