A Review of Deterministic Approaches to Stochastic Computing

Zhendong Lin
School of Microelectronics
Hefei University of Technology
Hefei, China
zdlin@mail.hfut.edu.cn

Guangjun Xie
School of Microelectronics
Hefei University of Technology
Hefei, China
gjxie8005@hfut.edu.cn

Shaowei Wang
School of Microelectronics
Hefei University of Technology
Hefei, China
2019010121@mail.hfut.edu.cn

Jie Han
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Canada
jhan8@ualberta.ca

Yongqiang Zhang
School of Microelectronics
Hefei University of Technology
Hefei, China
ahzhangyq@hfut.edu.cn

***Abstract*—Stochastic computing (SC) has emerged as an alternative to conventional computing with weighted binary representation. The operations in SC can be performed through simple logic gates to significantly reduce hardware complexity. The random bitstreams generated by stochastic number generators are exploited as the computing medium in SC. However, traditional operations in SC are inaccurate because of the inherent random fluctuations in bitstreams. To resolve this issue, deterministic approaches using the relatively prime stream length, rotation, and clock division of bitstreams, have been proposed for completely accurate computing. However, these approaches require much longer bitstreams, resulting in a longer computing latency and thus a larger energy consumption. For example, the bitstream length (BSL) is approximately 2^{2n} if two numbers with *n*-bit precision are multiplied using a deterministic approach. The studies aimed at lowering the latency and energy can be divided into two categories of serial and parallel designs to, respectively, reduce the BSL and parallelize bitstreams. These deterministic approaches to SC and the associated designs are reviewed in this paper with discussions of their strengths and weaknesses for possible improvements in future work.**

***Index Terms*—Stochastic computing, deterministic approaches, stochastic number generator.**

I. Introduction

In recent years, a variety of evolving technologies have emerged as potential alternatives to conventional weighted binary computation, including stochastic computing (SC). The main advantage of SC is that complex arithmetic operations can be implemented with compact and simple logic gates, hence achieving an ultra-low hardware complexity [1]. For example, the multiplication in SC can be performed by an AND gate [2]. Due to the inherent random fluctuations in bitstreams, SC is suitable for applications in which a slight inaccuracy is acceptable, such as image processing [2, 3], neural networks (NNs) [4, 5, 6, 7], and low-density parity check (LDPC) codes [8].

Deterministic approaches using relatively prime stream length (RPSL), rotation (Rot), and clock division (CloDiv) were first proposed in [9]. The SC operations are accurately performed by utilizing these deterministic approaches. In a deterministic approach, every bit in a bitstream interacts with every bit in the other one, so the random fluctuations in the bitstreams are eliminated. Therefore, the computing accuracy can be significantly improved with negligible additional hardware overhead, for example, for the Bernstein Polynomial circuit [9]. However, a key issue is that the bitstream length (BSL) for deterministic approaches exponentially increases. For example, the BSL is 2^{2n} when two *n*-bit numbers are multiplied if completely accurate results are required [10]. The advantages of deterministic approaches to SC will be offset because of the considerable energy consumption.

The studies aimed to alleviate this problem can be divided into two categories, one of which is to accelerate the serial computation and the other is to parallelize the bitstreams. The serial designs achieve acceleration by truncating bitstreams with an acceptable loss of accuracy. The parallel designs utilize parallel stochastic number generators (SNGs) to generate parallel bitstreams in only one clock cycle, hence lowering the computing latency and saving energy.

This paper is organized as follows. Section II introduces the basic components in SC. Section III presents the methods for accelerating computation in SC in detail. Section IV introduces the applications developed for the deterministic approaches. Section V reports the evaluation of different accelerators in terms of hardware cost and accuracy. Section VI describes the key challenges in deterministic approaches to SC. Section VII concludes this paper.

978-1-6654-0959-9/21/$31.00 @2021 IEEE

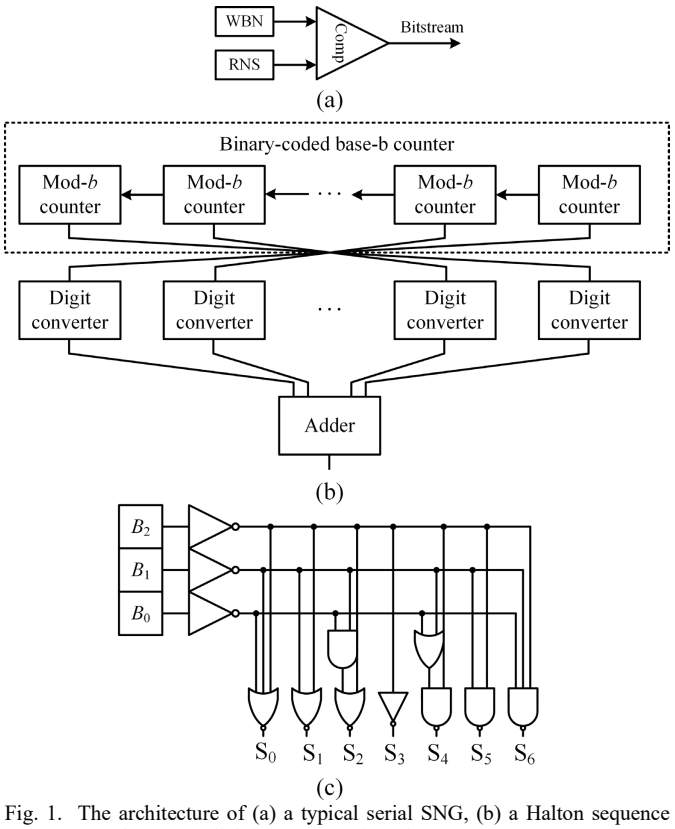

Fig. 1. The architecture of (a) a typical serial SNG, (b) a Halton sequence generator, and (c) a parallel thermometer-based SNG.

II. Basic Concepts

Various types of SNGs have been proposed for generating deterministic bitstreams since the SNG is an essential part in SC. This section illustrates the major architectures of SNGs and then presents the deterministic approaches in detail.

A. Stochastic Number Generators (SNGs)

SNGs can be divided into two main categories: serial SNGs and parallel ones. The probability or value encoded in a bitstream is obtained by simply calculating the proportion of 1s in it. For example, both 1100 and 10101010 represent 1/2. Fig. 1(a) shows the general structure of a serial SNG that consists of a random number source (RNS) and a binary comparator (Comp) [11]. Each bit in a bitstream is generated by comparing the pseudorandom number generated by the RNS with the weighted binary number (WBN). Generally, the RNS is based on a linear feedback shift register (LFSR) or binary counter, so the 0s and 1s in the generated bitstream are not uniformly distributed, leading to a lower computing accuracy.

Low-discrepancy (LD) sequences such as Halton and Sobol sequences have recently been used in the RNS to obtain a higher accuracy [12]. In a Halton sequence-based generator, 1s and 0s in the bitstreams are uniformly distributed, thereby producing better progressive precision [13]. Fig. 1(b) shows the Halton sequence-based RNS; its output is connected to the comparator in the SNG. Different mod-*b* counters are required for generating bitstreams with better uniform distributions of 1s and 0s. The sub-bitstreams generated by LD sequences-based SNGs are more accurate than those generated by the LFSR and counter-based ones, because they encode values closer to the expected ones. TABLE I lists the 3-bit precision bitstreams with the probability of 1/2 generated by SNGs with different RNS's. The sub-bitstreams of lengths 2 and 4 generated by a Halton-based SNG are completely accurate, while the counter and LFSR-based ones are not. This indicates that the bitstreams generated by the Halton-based SNG have the advantage of a fast convergence, so the computation can be terminated early to reduce latency.

TABLE I
The Bitstreams Gnerated by SNGs With Different RNS's

SNG	Bitstream
Counter-based	11110000 (1/2)
LFSR-based	10101100 (1/2)
Halton-based	10101010 (1/2)

Fig. 2. The deterministic approaches using: (a) relatively prime stream length, (b) rotation, and (c) clock division.

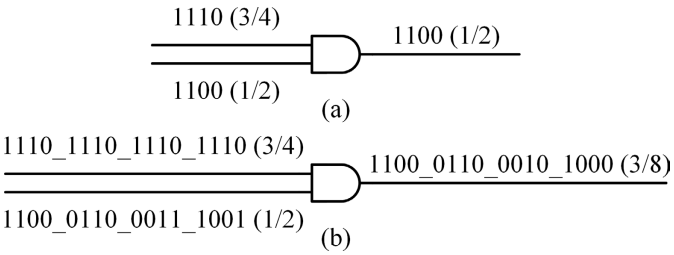

Fig. 3. The multiplication by (a) a traditional approach, and (b) a deterministic approach.

Using parallel SNGs, the latency can significantly be reduced because the parallel bitstreams are generated in only one clock cycle. An example is the thermometer-based SNG [14], as shown in Fig. 1(c). It only requires several logical gates to generate parallel bitstreams. However, few studies on parallel designs have been carried out because of the large hardware cost of massively parallel computing units. For example, it only requires one AND gate to perform multiplication of two n-bit numbers for a serial design, while this number increases to 2^n for a parallel design.

B. Deterministic Approaches

Fig. 2(a), (b), and (c) illustrate three different approaches, including RPSL, Rot, and CloDiv. For the RPSL approach, the lengths of two bitstreams are prime to each other. Two bitstreams are repeated several times to reach the same length. For the Rot approach, one bitstream is repeated while the other is rotated and repeated for the same number of times. For the

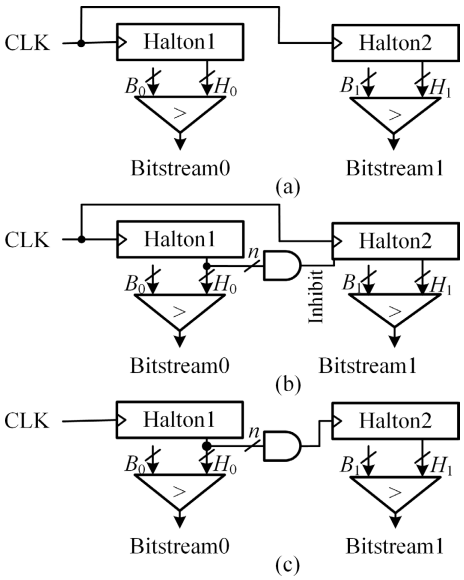

Fig. 4. The architectures of Halton-based deterministic SNGs using (a) Relatively prime stream length, (b) Rotation, and (c) Clock division.

CloDiv approach, two bitstreams are both repeated, while only one bit in one of them appears each time.

If completely accurate results are required, the BSL of deterministic bitstreams has to be 2^{2n} when multiplying two numbers with n-bit precision, which is much longer than that of traditional bitstreams. For example, Fig. 3 shows the multiplication of two 2-bit numbers using conventional SC and deterministic approaches. The output is completely accurate using the deterministic approach at the cost of an increased BSL and, thus, a long latency and low energy efficiency.

It seems that the RPSL approach produces the lowest accuracy since it truncates one of the bitstreams (b_3 is truncated, as shown in Fig. 2(a)), which results in a slight loss of accuracy. However, recent work has shown that it facilitates the design of time-based computation. This method translates analog signals into time-based signals represented by the duty ratio to perform computation in the SC domain. The RPSL approach is utilized to adjust the duty cycle to improve accuracy [15]. The computing latency can then be reduced to $1/2^n$ of that of the traditional SC. However, this method has several limitations. For example, it is hardly applicable to sequential circuits. For the Rot approach, in most cases, the outputs are more accurate because the input bitstreams show better progressive precision. Hence, it has a better performance in terms of accuracy than the RPSL and CloDiv approaches [16].

III. Accelerating Deterministic Approaches

To alleviate the latency issue, serial deterministic accelerators have been proposed to generate deterministic bitstreams with better progressive precision. Thus, a computation can be terminated in time at the cost of an acceptable loss of accuracy.

For parallel designs, the computing units and required stochastic-to-binary (S2B) converters are duplicated and parallelized, thus exacerbating the area and power cost of the circuits.

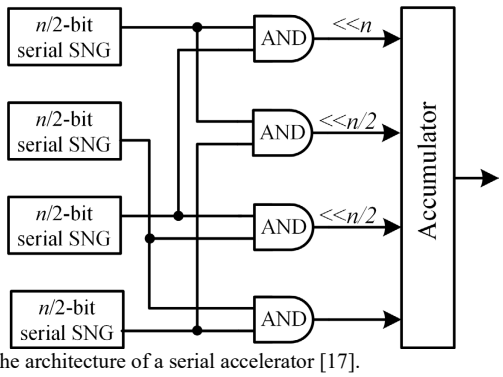

Fig. 5. The architecture of a serial accelerator [17].

A. Serial Deterministic Accelerators

In [9], a binary counter was exploited as an RNS. The generated bitstreams are sequences of consecutive 1s followed by sequences of 0s, thus leading to a low computing accuracy if a computation is terminated early. Different types of RNS's have been developed for generating bitstreams with better progressive precision. The RNS is replaced by an LFSR in [10], where 1s and 0s are randomly distributed, in contrast to counter-based SNGs. The latency is significantly reduced by terminating a computation in time if a slight inaccuracy is acceptable.

As mentioned above, LD sequences are utilized to improve accuracy, because they enable better progressive precision properties in generated bitstreams. Sobol sequence generators were exploited as RNS's to obtain a high accuracy in [17]. Furthermore, an accelerator was proposed for energy-efficient convolutional NNs by using deterministic bitstreams [18]. It achieves 70% savings in energy compared with conventional binary ones. In [16], Halton sequence-based deterministic SNGs were proposed for image processing. The latency is reduced by up to 128× compared with existing accelerators. Fig. 4 shows the architectures of Halton-based SNGs for generating three different deterministic bitstreams. If a slight inaccuracy is acceptable, the latency can be greatly reduced by terminating a computation in advance. Note that the two LD sequence-based solutions provide similar accuracy improvements, while the hardware cost of the Halton-based SNG is lower than that of a Sobol-based one [12].

An interesting solution referred to as bit-split was proposed in [19]. An n-bit binary number is split into K sub-numbers before converting it into bitstreams, where K is a power of 2. For example, an 8-bit binary number $B_7B_6B_5B_4B_3B_2B_1B_0$ is split into B_7B_6, B_5B_4, B_3B_2, and B_1B_0 when K is 4. Therefore, a complete bitstream is replaced by K sub-bitstreams. Then the K sub-bitstreams are fed into K^2 computing units and their outputs are left-shifted by different numbers of bits as the final results. This design shows excellent capabilities in reducing the computing latency. Fig. 5 shows the architecture of this accelerator when two n-bit numbers are multiplied (K=2, << indicates left-shift). However, the computing latency is reduced at the cost of increased design complexity and a larger area of the circuits, which offset its advantage.

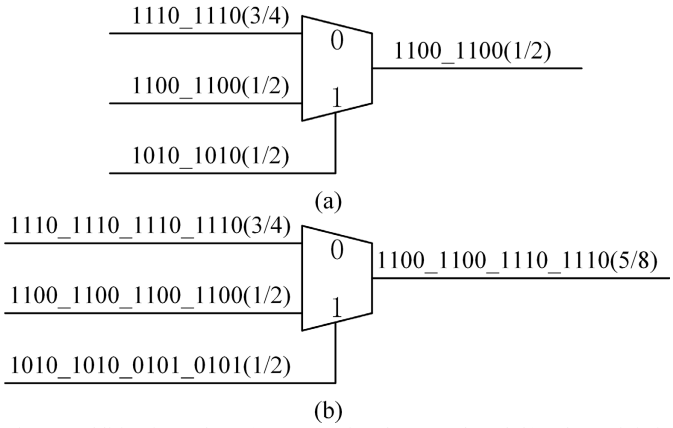

Fig. 6. Addition in SC by: (a) a conventional approach and (b) a deterministic approach.

B. Parallel Deterministic Accelerators

The hardware cost of computing units becomes considerably large using parallel datapaths. Although parallel designs are rarely studied because they are difficult to optimize, an energy-efficient NN accelerator that exploits the parallel thermometer-based SNG [14] was proposed in [20]. As an important component in an NN, a multiply accumulate (MAC) unit is accelerated by using this parallel SNG with the Rot approach, resulting in a huge energy reduction. However, the area becomes much larger because of the parallel computing units and S2B converters (about 9× that of the serial designs).

A parallel deterministic accelerator using shuffling and sub-sampling networks was proposed in [21]. The BSL is reduced to $1/2^n$ of that of conventional deterministic approaches for this accelerator with n-bit precision. Compared with conventional binary implementations, the area-delay product (ADP) is improved by 10×. However, a serious drawback for this work is that the accuracy is significantly reduced, which limits its application.

IV. Applications

This section discusses several applications of the deterministic approaches.

A. Multiplication and Addition

Multiplication and addition are two basic and important operations in SC.

As shown in Fig. 3, the output bitstream encodes the accurate result without any random fluctuations. Here, the stochastic computing correlation (SCC) defined in (1) between the two input bitstreams is reduced to 0 [22]. For correlation-sensitive circuits, such as the multiplier, a higher accuracy can be obtained with a lower absolute value of SCC between the input bitstreams [23].

$$SCC(X,Y)=\begin{cases}\dfrac{p_{X\cap Y}-p_Xp_Y}{\min(p_X,p_Y)-p_Xp_Y}, & p_{X\cap Y}-p_Xp_Y>0\\ 0, & p_{X\cap Y}-p_Xp_Y=0\\ \dfrac{p_{X\cap Y}-p_Xp_Y}{p_Xp_Y-\max(p_X+p_Y-1,0)}, & p_{X\cap Y}-p_Xp_Y<0\end{cases} \quad (1)$$

The authors in [16] suggest utilizing Halton-based SNGs for accelerating the addition in SC. The computation is completely accurate with a BSL of 2^{n+1} when two n-bit numbers are summed. For example, Fig. 6 compares the addition of two numbers of 3-bit precision with and without using deterministic approaches. The output bitstream is completely accurate with the probability 5/8, due to the better progressive precision property of Halton sequences than the traditional approach. Note that the correlation of two input signals does not affect computing accuracy, therefore they are simply repeated. The deterministic approach (Rot in this case) is only applied in the select signal of the multiplexer [24].

B. Image Processing

Conventional SC has extensively been applied in various image processing algorithms. One of the key issues is how to improve its accuracy at low energy consumption. For example, the accuracy is significantly improved by deterministic approaches in [16]. The BSL can be reduced to a very short length if a slight inaccuracy is acceptable (e.g., the mean absolute error is lower than 0.1%). The detailed comparison will be shown in the next subsection to illustrate the advantage of deterministic accelerators.

C. Neural Networks (NNs)

NNs consisting of a large number of MAC units are an emerging application for SC. The energy consumption can significantly be reduced by accelerating the MAC operations. For example, in [18], an NN is designed with an LD sequence-based serial accelerator for saving energy. In the parallel deterministic accelerator proposed in [20], the latency is significantly reduced at the cost of an increased area, thereby saving energy.

V. Comparison

Since the parallel designs are rarely studied, this section focuses on the serial designs. Different types of serial SNGs for generating deterministic bitstreams are first reviewed. Then, their performance in terms of hardware usage and computing accuracy are compared through the application of an image processing algorithm.

A. Hardware

The deterministic bitstreams are generated by using different types of SNGs, hence the hardware measurements are compared first, as shown in TABLE II, for the SNGs in [9], [10], and [16] in terms of area, power, critical path delay (CPD), area-delay product (ADP), power-delay product (PDP), and energy-delay product (EDP). All these implementations are with 8-bit precision, synthesized by the Synopsys Design Compiler with TSMC's 45 nm library. As can be seen, the hardware cost of the LFSR-based SNG is the highest. The measurements are similar for the counter-based and Halton-based SNGs. In fact, the base-2 Halton sequence generators can be utilized to significantly reduce the hardware cost.

B. Accuracy

Robert's cross edge detector [25] has been considered for

TABLE II
THE HARDWARE COMPARISON OF DIFFERENT TYPES OF SNGS

Approach	SNG	Area (μm^2)	Power (μW)	Delay (ns)	ADP ($\mu m^2 \times ns$)	PDP ($pJ \times 10^{-3}$)	EDP ($10^{-24} Js$)
Counter [9]	RPSL	156.67	1.89	2.26	354.08	4.26	9.63
	Rot	151.89	1.89	2.54	385.79	4.78	12.13
	CloDiv	154.01	1.88	2.54	391.20	4.77	12.10
LFSR [10]	RPSL	193.65	2.36	2.43	470.56	5.73	13.92
	Rot	192.58	2.35	2.43	467.98	5.71	13.88
	CloDiv	192.58	2.35	2.43	467.98	5.71	13.88
Haton [16]	RPSL	156.14	1.90	2.24	349.76	4.25	9.52
	Rot	154.28	1.91	2.24	345.59	4.29	9.61
	CloDiv	153.75	1.90	2.24	344.40	4.25	9.52

TABLE III
THE MAE (%) OF DIFFERENT SNGS FOR IMPLEMENTATIONS OF THE ROBERT CROSS EDGE DETECTOR VERSUS DIFFERENT BSLS

Approach	SNG	2^{16}	2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^{9}	2^{8}
Counter [9]	RPSL	**0.0969**	1.0600	1.5700	1.7800	1.8500	1.8800	1.8900	1.9000	1.9000
	Rot	**0.0957**	1.0600	1.6400	1.8200	1.8700	1.8800	1.8900	1.8900	1.8900
	CloDiv	**0.0957**	2.2100	2.2100	2.2100	2.2100	2.2100	2.2100	2.2100	2.2100
LFSR [10]	RPSL	0.0966	**0.1000**	0.1100	0.1300	0.1500	0.1800	0.2300	0.3000	0.4000
	Rot	0.0957	**0.1000**	0.1100	0.1300	0.1500	0.1800	0.2300	0.3000	0.4000
	CloDiv	**0.0957**	0.1700	0.2500	0.3200	0.4600	0.6300	0.8700	1.1800	2.2100
Haton [16]	RPSL	0.0969	0.0969	0.0969	0.0970	0.0973	0.0980	0.1000	**0.1000**	1.9000
	Rot	0.0957	0.0957	0.0957	0.0957	0.0957	0.0957	0.0957	**0.0957**	1.8900
	CloDiv	0.0957	0.0957	0.0957	0.0957	0.0957	0.0957	0.0957	**0.0957**	2.2100

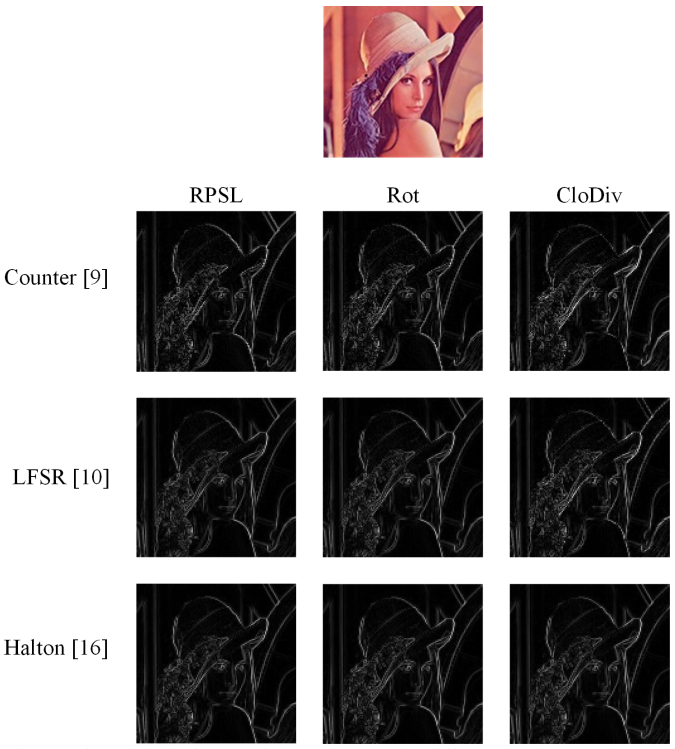

Fig. 7. The original photo and output images by different implementations of the edge detector when the BSL is 2^9.

verifying the performance of different SNGs in terms of computing accuracy using the mean absolute error (MAE). A photo with 128×128 pixels is used as an input image/data for these circuits with 8-bit precision. The experimental results are shown in TABLE III. The MAE increases as the BSL decreases.

It is obvious that the state-of-the-art Halton-based design outperforms the others. If a slight inaccuracy (lower than 0.1%) is acceptable, it performs up to 128× faster than the other designs, as marked in red in the table. Fig. 7 shows the original photo and the comparison of different implementations when the BSL is 2^9. The difference in the images is not obvious because it is difficult for human eyes to recognize a difference lower than 5% between two similar images [26]. Nevertheless, the latency is still long even with the significant reduction, compared with that obtained by conventional binary designs.

VI. CHALLENGES

Deterministic approaches to SC are effective for achieving highly accurate computation. However, this advantage is offset by some shortcomings.

A. Serial Design

The BSL exponentially increases by using the deterministic approaches. The main challenge is how to reduce the BSL with an acceptable loss of other performance metrics. The bit-split method has been proposed for reducing the BSL. Unfortunately, the complexity of computing units is also significantly increased. LD sequence-based SNGs have been developed for accelerating computation and reducing the latency. However, the latency is still much longer than conventional binary designs.

B. Parallel Design

Parallel SNGs provide a potential solution for accelerating deterministic computing in SC. The latency is reduced to only one clock cycle for generating the parallel bitstreams. However, the huge hardware cost is one of its main disadvantages. This imposes a limitation on applications that require a low area cost. It is therefore critical to reduce the complexity of parallel computing units. Using the sub-sampling method, however, the computing accuracy considerably declines with the reduction of hardware complexity. Therefore reducing the design

complexity of parallel SNGs at a reasonable cost will need to be addressed in future work.

VII. Conclusion

Conventional SC circuits are inaccurate because of the inherent random fluctuations in the bitstreams generated by SNGs. Deterministic approaches have been proposed for eliminating the fluctuations and successfully exploited in various practical applications, including image processing and NNs. However, a significant drawback is the long BSL that results in low energy efficiency. Two main solutions, i.e., serial and parallel designs, have recently been employed to alleviate this issue.

Several serial designs, such as the bit-split, time-based, and LD sequence-based methods, have been proposed for accelerating deterministic computing and saving energy. The bit-split method reduces the latency, while the design complexity substantially increases. The latency is significantly reduced by the time-based method. However, it is hard to apply in practice because it is limited to only the first level of logic in a circuit. An LD sequence generator with a better progressive precision property appears to be more effective for accelerating serial deterministic computing because the latency can be significantly reduced with negligible hardware overhead, especially when a slight loss of accuracy is acceptable.

Few studies have been carried out for parallel deterministic designs due to the large areas of parallel computing units. Nevertheless, parallel designs may provide a feasible solution for a deterministic computing accelerator because the latency is reduced to only one clock cycle by using parallel bitstreams. Future research is needed to reduce the design complexity of parallel deterministic computing units to save hardware overhead.

References

[1] A. Alaghi, W. Qian, and J. Hayes, "The promise and challenge of stochastic computing," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.,* vol. 37, no. 8, pp. 1515-1531, Aug. 2018.

[2] B. Gaines, "Stochastic computing systems," *Advances in information systems science*, Advances in information systems science J. T. Tou, ed., pp. 37-172: Springer, Boston, MA, 1969.

[3] P. Li, and D. Lilja, "Using stochastic computing to implement digital image processing algorithms," in the 2011 IEEE 29th International Conference on Computer Design (ICCD), Amherst, MA, USA, 2011, pp. 154-161.

[4] Y. Liu, S. Liu, Y. Wang, F. Lombardi, and J. Han, "A survey of stochastic computing neural networks for machine learning applications," *IEEE Trans. Neural Netw. Learn. Syst.,* vol. 32, no. 7, pp. 2809 - 2824, Jul. 2021.

[5] V. Lee, A. Alaghi, J. Hayes, V. Sathe, and L. Ceze, "Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing," in the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, Lausanne, Switzerland, 2017, pp. 13-18.

[6] A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. 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.

[7] Y. Xie, S. Liao, B. Yuan, Y. Wang, and Z. Wang, "Fully-parallel area-efficient deep neural network design using stochastic computing," *IEEE Trans. Circuits Syst. II Express Briefs,* vol. 64, no. 12, pp. 1382-1386, Dec. 2017.

[8] V. C. Gaudet, and A. C. Rapley, "Iterative decoding using stochastic computation," *Electron. Lett.,* vol. 39, no. 3, pp. 299-301, Feb. 2003.

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

[10] H. Najafi, and D. Lilja, "High quality down-sampling for deterministic approaches to stochastic computing," *IEEE Trans. Emerging Top. Comput.,* vol. 9, no. 1, pp. 7-14, Mar. 2021.

[11] T. Chen, P. Ting, and J. Hayes, "Achieving progressive precision in stochastic computing," in the 2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Montreal, QC, Canada, 2017, pp. 1320-1324.

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

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

[14] Y. Zhang, R. Wang, X. Zhang, Z. Zhang, J. Song, Z. Zhang, Y. Wang, and R. Huang, "A parallel bitstream generator for stochastic computing," in the 2019 Silicon Nanoelectronics Workshop (SNW), Kyoto, Japan, 2019, pp. 1-2.

[15] M. Najafi, S. Jamali-Zavareh, D. Lilja, M. Riedel, K. Bazargan, and R. Harjani, "An overview of time-based computing with stochastic constructs," *IEEE Micro,* vol. 37, no. 6, pp. 62-71, Nov.-Dec. 2017.

[16] Z. Lin, G. Xie, W. Xu, J. Han, and Y. Zhang, "Accelerating stochastic computing using deterministic halton sequences," *IEEE Trans. Circuits Syst. II Express Briefs,* vol. 68, no. 10, pp. 3351-3355, Oct. 2021.

[17] M. Najafi, D. Lilja, and M. Riedel, "Deterministic methods for stochastic computing using low-discrepancy sequences," in the 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Diego, CA, USA, 2018, pp. 1-8.

[18] S. Faraji, M. Najafi, B. Li, D. Lilja, and K. Bazargan, "Energy-efficient convolutional neural networks with deterministic bit-stream processing," in the 2019 Design, Automation & Test in Europe Conference & Exhibition, Florence, Italy, 2019, pp. 1757-1762.

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

[20] Y. Zhang, R. Wang, X. Zhang, Y. Wang, and R. Huang, "Parallel hybrid stochastic-binary-based neural network accelerators," *IEEE Trans. Circuits Syst. II Express Briefs,* vol. 67, no. 12, pp. 3387 - 3391, Dec. 2020.

[21] Z. Wang, S. Mohajer, and K. Bazargan, "Low latency parallel implementation of traditionally-called stochastic circuits using deterministic shuffling networks," in the 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), Jeju, Korea (South), 2018, pp. 337-342.

[22] H. Abdellatef, M. Khalil-Hani, and N. Shaikh-Husin, "Accurate and compact stochastic computations by exploiting correlation," *Turk. J Electr. Eng. Comput. Sci.,* vol. 27, no. 1, pp. 547-564, 2019.

[23] V. Lee, A. Alaghi, and L. Ceze, "Correlation manipulating circuits for stochastic computing," in the 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 2018, pp. 1417-1422.

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

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

[26] P. Li, and D. Lilja, "Accelerating the performance of stochastic encoding-based computations by sharing bits in consecutive bit streams," in the Proceedings of the 2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors, New York, 2013, pp. 257-260.