

## Research Article

# Random Error Reduction Scheme for Combinational Stochastic Circuit

**Ye Cheng and Jianhao Hu**

*National Key Lab. of Communications, University of Electronic Science and Technology of China, Chengdu 611731, China*

Correspondence should be addressed to Ye Cheng; [chengye\\_aobao@163.com](mailto:chengye_aobao@163.com)

Received 22 November 2016; Revised 15 March 2017; Accepted 11 April 2017; Published 8 October 2017

Academic Editor: Paolo Crippa

Copyright © 2017 Ye Cheng and Jianhao Hu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In conventional stochastic computation, all the input streams are Bernoulli sequences (BSs), which may result in large random error. To reduce random error and improve computational accuracy, some other sequences have been reported as alternatives to BSs. However, these sequences only apply to the specific stochastic circuits, have difficulties in hardware generation, or have length constraints. To this end, new sequences without these disadvantages should be considered. This paper proposes the random error analysis method for stochastic computation based on autocorrelation sequence (AS), which is more general than the conventional one based on BS. The analysis results show that we can use the proper ASs as input streams of stochastic circuits to reduce random error. On the basis of that conclusion, we propose the random error reduction scheme based on maximal concentrated autocorrelation sequence (MCAS) and BS, both of which are ASs. MCAS and BS are applicable to any combinational stochastic circuit, are easily generated by hardware, and have no length constraints, which avoid the disadvantages of sequences in the previous work. Moreover, we apply the proposed random error reduction scheme into several typical stochastic circuits as case studies. The simulation results confirm the effectiveness of the proposed scheme.

## 1. Introduction

Stochastic computation, first introduced in 1960s [1], has been widely used in many fields: neural network [2, 3], digital image processing [4–7], channel decoding [8–10], MIMO detection [11, 12], reliability evaluation [13], and so on. The main idea of stochastic computation is to represent continuous values with stochastic bit streams, which makes it possible to perform complex arithmetic computations with simple bitwise operations. One obvious advantage of stochastic computation is that the hardware implementation of the stochastic arithmetic is much simpler than the traditional arithmetic. For example, as shown in Figure 1, multiplication and scaling addition can be performed with an AND gate and a multiplexer, respectively. The simplicity of hardware also brings short critical path and high clock rate. Moreover, all the bits in a stochastic bit stream have the same significance. That means the stochastic arithmetic is unary coding arithmetic which has better fault tolerance than the traditional binary arithmetic. Recently, skew tolerance has also been introduced as another inherent advantage of stochastic computation.

Despite the above advantages, a disadvantage of stochastic computation is that the random fluctuations of stochastic bit streams introduce the random error, which causes the computation result to be a random variable. In [4], it is suggested that, in most cases, the random error will dominate the other errors such as quantization error and approximate error. Thus, the random error should be carefully investigated and addressed in stochastic computation area.

Thus far, many statistical models for random error analysis have been proposed. They are all based on variance evaluations. In the early work, it is assumed that all the input streams of stochastic circuits are BSs. Qian et al. [4] propose the variance evaluation model base on BSs. However, in real application, the random error for stochastic arithmetic based on BSs is often too large. Therefore, people have made efforts to reduce random error by using other sequences as input streams of stochastic circuits.

Braendler et al. [3] employ two deterministic bit streams (DBSs) as input streams for the stochastic circuit AND gate. The output stream is also deterministic, which means there are no random fluctuations. Although it eliminates random

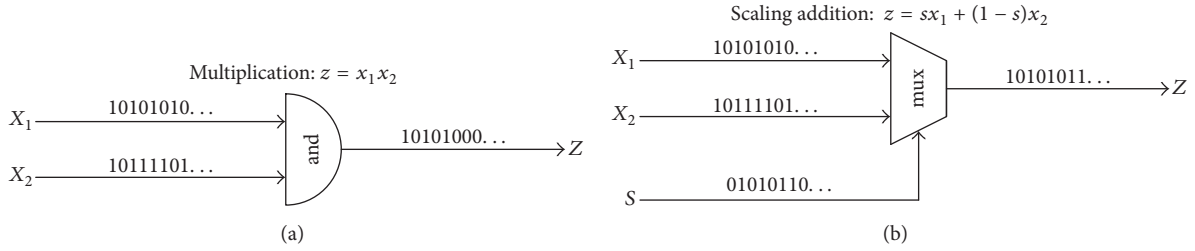


FIGURE 1: Stochastic circuit of arithmetic. (a) Multiplication; (b) scaling addition.

fluctuations and improves computational accuracy, the use of these DBSs is narrowly restricted to the single AND gate. They can not apply to more complex stochastic circuits.

Alaghi and Hayes [14] propose a general framework for analyzing stochastic circuits with correlated inputs. They demonstrate that using correlated inputs can significantly reduce the random error for some stochastic circuits. However, the use of correlated inputs has its limitation. For example, as illustrated in [15], if the two input streams applied to an AND gate are correlated, say, each has the same bit-pattern of value  $x$ , then the output will also have value  $x$  instead of  $x^2$ . Hence, the AND gate will not perform multiplication if its inputs are correlated. Similarly, many other commonly used stochastic computational elements can not perform the expected arithmetic with correlated inputs, which largely restricts their applications in stochastic computation systems.

Han et al. [13] use fixed-ones random-permutation sequences (FRSs) as input streams for stochastic circuits. They present the variance evaluation model for the basic stochastic computational elements, including AND gate and invert. Furthermore, they prove that the use of FRSs as input streams reduces random error for stochastic computation compared to the conventional use of BSs. FRSs can be widely used in stochastic arithmetic. They apply to any combinational stochastic circuit. In [13], FRSs are generated by software and used as input streams of the stochastic circuit for reliability evaluation. However, the low cost hardware generation of FRS is difficult to implement. Thus, FRSs are not suitable for stochastic circuits with input streams generated by hardware.

Najafi et al. [16] use pulse-width modulation (PWM) to time-encode values for more efficient stochastic computing. They represent values by periodic PWM signals. If two PWM signals are not harmonically related, stochastic multiplication of numbers represented by them obtains the best accuracy when running the operation for the least common multiple (LCM) or multiples of the LCM of the period of the inputs. For instance,  $X = 3/5$  is represented by a PWM signal with period of 5, namely, 1, 1, 0, 0, 0, and  $Y = 1/3$  is represented by a PWM signal with period of 3, namely, 1, 0, 0. The LCM of the period of  $X$  and  $Y$  is 15. Thus, when multiplying  $X$  and  $Y$  with an AND gate, let both of the two bit streams run for 15 clock cycles; that is,  $X = 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0$ , and  $Y = 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0$ . Taking the bitwise AND of  $X$  and  $Y$ , the output stream is  $Z = 1, 0, 0, 0, 0, 0, 0$ ,

1, 0, 0, 0, 0, 0, 0, 0, 0, which represents  $2/15 = 2/5 \times 1/3$ . However, if the operation runs for only 10 clocks, that is,  $X = 1, 1, 0, 0, 0, 1, 1, 0, 0, 0$ , and  $Y = 1, 0, 0, 1, 0, 0, 1, 0, 0, 1$ , the output stream is  $Z = 1, 0, 0, 0, 0, 0, 1, 0, 0, 0$ , which represents  $2/10$ , not the expected value. Similarly, some other stochastic arithmetic, using PWM signals as input streams, can get the expected results only if the operations run for the required time. Hence, we can see that there are length constraints on stochastic computation based on PWM signals. The length of input streams is not flexible but determined by the period of input streams.

From above, we can see that the main issues of stochastic arithmetic based on different input streams are as follows:

- (i) BSs: large random error;
- (ii) DBSs or correlated inputs: only applicable for the specific stochastic circuits;
- (iii) FRSs: difficulties in hardware generation;
- (iv) PWM signals: length constraints.

To avoid these disadvantages, we need to use other proper sequences as input streams. Considering the autocorrelation between different bits of a sequence, we propose the notion of AS. And then we analyze the random error of stochastic arithmetic on the assumption that input streams are ASs. By definition, BS can be seen as a special AS with equivalent autocorrelation coefficients. Thus, the proposed random error analysis method based on AS is more general than the conventional one based on BS.

According to our analysis results, if proper ASs are used as input streams, the random error can be reduced. Based on that conclusion, we investigate how to reduce random error for any combinational stochastic circuit by employing MCASs and BSs, both of which are ASs, as input streams. Then, We adopt several typical stochastic circuits as case studies and find that the use of MCASs and BSs can significantly improve the calculation performance. MCAS has no difficulties in hardware generation and the hardware cost of the MCAS generator is merely the same as that of the conventional BS generator. Moreover, from the simulation results, the random error reduction scheme based on MCAS and BS is valid for any input stream length and has no length constraints. Hence, by using MCASs and BSs as input streams, we avoid the disadvantages in the previous work and achieve the following goals:

- (i) Random error reduction: the random error is reduced compared to using only BSs as input streams;
- (ii) Wide applications: the random error reduction scheme based on MCAS and BS is applicable to any combinational stochastic circuit;
- (iii) Easy hardware generation: the MCAS and BS generator are easily implemented by hardware;
- (iv) Flexible length: the MCAS and BS have no length constraints.

Parts of this paper are based on our prior work entitled "Random Error Analysis and Reduction for Stochastic Computation Based on Autocorrelation Sequence," which has been presented at the IEEE International Symposium on Circuits and Systems (ISCAS), 2014. In the conference paper, we briefly present a general random error analysis method for stochastic computation and illustrate how to reduce random error for a typical stochastic circuit. Compared with the conference version, this journal manuscript has more new contents updated and more insightful discussion, which are stated as follows:

- (i) Detailed illustration of the general random error analysis method is provided.
- (ii) The random error reduction scheme for any combination stochastic circuit is proposed.
- (iii) The complete mathematical proof for the effectiveness of the proposed random error reduction scheme is given.
- (iv) More typical stochastic circuits are presented as case studies.

The rest of the paper is structured as follows: Section 2 analyzes the random error based on AS. Section 3 illustrates the random error reduction scheme based on MCAS and BS for any combinational stochastic circuit. Section 4 presents several typical case studies. Section 5 gives simulation results. Finally, we give conclusions and future directions of research.

## 2. Random Error Analysis Method

**2.1. Expectation and Variance.** In stochastic computation, a stochastic bit stream  $X$  with length of  $n$  can be written as

$$X = [X(1), X(2), \dots, X(n)], \quad (1)$$

$$X(i) \in \{0, 1\}, \quad i = 1, 2, \dots, n.$$

The expectation of  $X$  is defined by

$$E_X = E\left(\frac{1}{n} \sum_{i=1}^n X(i)\right) = \frac{1}{n} \sum_{i=1}^n E(X(i)), \quad (2)$$

and the variance of  $X$  is defined by

$$\text{Var}_X = \text{Var}\left(\frac{1}{n} \sum_{i=1}^n X(i)\right) = \frac{1}{n^2} \text{Var}\left(\sum_{i=1}^n X(i)\right). \quad (3)$$

Consider an arithmetic operation  $F$  with  $k$  normalized input numbers  $x_1, x_2, \dots, x_k$  and a result  $z$ , which can be expressed by  $z = F(x_1, x_2, \dots, x_k)$ . In the stochastic implementing of  $F$ , each input number  $x_r, 1 \leq r \leq k$ , is represented by the corresponding stochastic bit stream, denoted by  $X_r$ , where  $E_{X_r} = x_r$ . We employ these stochastic bit streams as input streams of stochastic circuit. Then, the stochastic computational result, denoted by  $z'$ , can be obtained from the output stream  $Z$  as follows:

$$z' = \frac{1}{n} \sum_{i=1}^n Z(i). \quad (4)$$

There are two important statistics for stochastic arithmetic. The first one is the expectation of output stream. Since  $z'$  should be an unbiased estimator of the actual result  $z$ , we have

$$E(z') = z = F(x_1, x_2, \dots, x_k). \quad (5)$$

From (2) and (4), (5) can be rewritten as  $E_Z = z = F(x_1, x_2, \dots, x_k)$ . Thus, the expectation of output stream indicates the arithmetic operation performed by stochastic circuit.

Another important statistic is the variance of output stream. Considering that  $z'$  is an unbiased estimator of the actual result  $z$ , random error can be calculated in the form of mean square error (MSE) as

$$\begin{aligned} \text{MSE} &= E\left((z' - z)^2\right) = E\left((z' - E(z'))^2\right) \\ &= \text{Var}(z'). \end{aligned} \quad (6)$$

By (3) and (6),  $\text{MSE} = \text{Var}_Z$ . Therefore, the random error for stochastic computation can be measured by the variance of output stream.

**2.2. Autocorrelation Sequence.** For a bit stream  $X$  with length of  $n$ , the correlation between two different bits can be measured by autocorrelation coefficient, which is defined by  $R_{ij}^X = E(X(i)X(j)), 1 \leq i < j \leq n$ . For example, if  $X$  is a BS, its autocorrelation coefficients are  $R_{ij}^X = (E_X)^2, 1 \leq i < j \leq n$ ; that is, all the autocorrelation coefficients of BS are equivalent to the square of its expectation.

In the practical applications, input streams may have a variety of autocorrelation coefficients. Considering that, we propose AS, whose autocorrelation coefficients are not necessarily equivalent to each other.

**Definition 1.** A bit stream  $X$ , which represents the probability  $x$ , is called an AS on the following condition:  $E_X = x$ .

From Definition 1, AS is more general compared to BS and BS can be seen as special case of AS whose autocorrelation coefficients are equivalent to the square of its expectation.

**2.3. Random Error Analysis Based on AS.** For clarity, we define the following notations: let BI denote the case that

all the input streams are BSs; let AI denote the case that all the input streams are ASs; let  $E(\text{BI})$  denote the expectation of output stream  $Z$  in the case of BI; let  $E(\text{AI})$  denote the expectation of output stream  $Z$  in the case of AI; let  $\text{Var}(\text{BI})$  denote the variance of output stream  $Z$  in the case of BI; let  $\text{Var}(\text{AI})$  denote the variance of output stream  $Z$  in the case of AI.

For any stochastic circuit, we expect that the arithmetic computed in the case of AI is the same as that in the case of BI. As illustrated in Section 2.1, the expectation of output stream indicates the arithmetic operation performed by stochastic circuit. Hence, the expectations of output stream should be equivalent to each other in the two cases and we have

$$E(\text{AI}) = E(\text{BI}). \quad (7)$$

Equation (7) is the essential condition for the random error analysis in the case of AI. On the premise of (7), the random error base on AS can be measured by the variance of the output stream  $Z$  as follows:

$$\begin{aligned} \text{Var}(\text{AI}) &= \frac{1}{n^2} \text{Var} \left( \sum_{i=1}^n Z(i) \right) \\ &= \frac{1}{n^2} \sum_{i=1}^n \sum_{j \neq i}^n (E(Z(i)Z(j)) - E(Z(i))E(Z(j))) \\ &\quad + \frac{1}{n^2} \sum_{i=1}^n \text{Var}(Z(i)) \\ &= \frac{1}{n} E(\text{BI}) - E^2(\text{BI}) + \frac{1}{n^2} \sum_{i=1}^n E(Z(i)Z(i)). \end{aligned} \quad (8)$$

*Example 2.* As shown in Figure 1, two stochastic bit streams  $X_1$  and  $X_2$  are employed as the input streams for an AND gate. When  $X_1$  and  $X_2$  are BSs,  $E(X_1(i)) = E_{X_1}$ ,  $E(X_2(i)) = E_{X_2}$ ,  $i = 1, 2, 3, \dots, n$ . The expectation of output streams can be calculated as

$$\begin{aligned} E(\text{BI}) &= \frac{1}{n} \sum_{i=1}^n E(X_1(i)X_2(i)) = \frac{1}{n} \sum_{i=1}^n E_{X_1} E_{X_2} \\ &= E_{X_1} E_{X_2}. \end{aligned} \quad (9)$$

Now, consider that the input streams  $X_1$  and  $X_2$  are ASs with the following properties:  $E(X_1(i)) = E_{X_1}$  or  $E(X_2(i)) = E_{X_2}$ ,  $i = 1, 2, 3, \dots, n$ . Without losing generality, we suppose that  $E(X_1(i)) = E_{X_1}$ . The expectation of output stream can be calculated as

$$\begin{aligned} E(\text{AI}) &= \frac{1}{n} \sum_{i=1}^n E(X_1(i)X_2(i)) = \frac{1}{n} E_{X_1} \sum_{i=1}^n E(X_2(i)) \\ &= E_{X_1} E_{X_2}. \end{aligned} \quad (10)$$

By (9) and (10), (7) is satisfied. Thus, the random error can be calculated by (8) as follows:

$$\text{Var}(\text{AI}) = \frac{1}{n} E_{X_1} E_{X_2} - E_{X_1}^2 E_{X_2}^2 + \frac{1}{n^2} \sum_{i=1}^n \sum_{j \neq i}^n R_{ij}^{X_1} R_{ij}^{X_2}. \quad (11)$$

Hence, the variance of output stream in the case of AI is determined by the expectations and autocorrelation coefficients of input streams.

*2.4. AI Affection Evaluation.* In order to evaluate how much AI affects the random error, we propose differential variance, denoted by DV, where  $DV = \text{Var}(\text{AI}) - \text{Var}(\text{BI})$ . As suggested in [4], the variance of output stream in the case of BI can be calculated by

$$\text{Var}(\text{BI}) = \frac{E(\text{BI})(1 - E(\text{BI}))}{n}. \quad (12)$$

Then we have the following theorem.

**Theorem 3.** *The range of DV is as follows:*

$$\frac{E^2(\text{BI}) - E(\text{BI})}{n} \leq DV \leq \frac{n-1}{n} (E(\text{BI}) - E^2(\text{BI})). \quad (13)$$

*Proof.* See our prior work [17].  $\square$

According to Theorem 3, the upper bound of DV is nonnegative and the lower bound of DV is nonpositive. Thus, DV can be positive, which indicates that the improper use of AS will result in the increase of random error. On the other hand, DV can be negative, which indicates that the proper use of AS will reduce random error. Therefore, we get the following conclusions.

- (i) Since the random error in the case of AI can be less than that in the case of BI, the random error reduction scheme based on AS is feasible.
- (ii) Since the random error in the case of AI can be greater than that in the case of BI, it is important to select proper AS to avoid this situation.

### 3. Random Error Reduction Scheme

In the last section, we find that the proper use of AS is able to reduce the random error for stochastic computation. On the basis of that conclusion, we will discuss how to employ ASs as input streams for the combinational stochastic circuit to reduce the random error.

*3.1. Combinational Stochastic Circuit.* For any combinational stochastic circuit, the relationship between input streams and output stream can be expressed by

$$Z(i) = \sum_{l_1 l_2 \dots l_k} A_{l_1 l_2 \dots l_k} X_1(i)^{l_1} X_2(i)^{l_2} \dots X_k(i)^{l_k}, \quad (14)$$

where  $A_{l_1, k}$  is a constant integer and  $l_r \in \{0, 1\}$ ,  $1 \leq r \leq k$ . For convenience of illustration, we denote  $l_{1, k}$  by the subscript  $l_1 l_2 \dots l_k$ . The formulation  $A_{l_1, k} X_1(i)^{l_1} X_2(i)^{l_2} \dots X_k(i)^{l_k}$ , where  $A_{l_1, k} \neq 0$ , is called a product term and denoted by  $P_{l_1, k}^i$ . Then, (14) can be rewritten as  $Z(i) = \sum_{l_{1, k} \in S_p} P_{l_{1, k}}^i$ , where  $S_p = \{l_{1, k} : A_{l_{1, k}} \neq 0\}$ . For the product term  $P_{l_{1, k}}^i$ , if  $l_r = 1$ ,  $X_r$  is called a

factor of  $P_{l_{1,k}}^i$ . Let  $M(r)$  denote the set of subscripts of product terms that have  $X_r$  as a factor; that is,  $M(r) = \{l_{1,k} : l_r = 1, l_{1,k} \in S_p\}$ . Similarly, let  $N(l_{1,k})$  denote the set of subscripts of input streams that are factors of  $P_{l_{1,k}}^i$ ; that is,  $N(l_{1,k}) = \{r : l_r = 1, 1 \leq r \leq k\}$ .

**Definition 4.** An input stream  $X_r$  is called a positive input stream (PIS), if it satisfies

$$\frac{\sum_{l_{1,k} \in M(r)} P_{l_{1,k}}^i}{X_r(i)} \geq 0, \quad (15)$$

$$\forall X_1(i), X_2(i), \dots, X_{r-1}(i), X_{r+1}(i), \dots, X_k(i).$$

The set of subscripts of PISs in stochastic circuit is denoted by  $S_{\text{pis}}$ .

**Example 5.** Consider the stochastic circuit with Boolean function as follows:  $Z(i) = X_1(i) \wedge X_2(i) \vee X_3(i)$ , and we have

$$\begin{aligned} Z(i) &= X_1(i) X_2(i) (1 - X_3(i)) + X_1(i) X_2(i) X_3(i) \\ &\quad + X_1(i) (1 - X_2(i)) X_3(i) \\ &\quad + (1 - X_1(i)) X_2(i) X_3(i) \\ &\quad + (1 - X_1(i)) (1 - X_2(i)) X_3(i), \end{aligned} \quad (16)$$

which can be rewritten in the form of (14) as

$$Z(i) = X_1(i) X_2(i) + X_3(i) - X_1(i) X_2(i) X_3(i). \quad (17)$$

There are three product terms and  $Z(i) = P_{110}^i + P_{001}^i - P_{111}^i$ .  $X_1$  is a factor of  $P_{110}^i$  and  $P_{111}^i$ ,  $X_2$  is a factor of  $P_{110}^i$  and  $P_{111}^i$ , and  $X_3$  is a factor of  $P_{001}^i$  and  $P_{111}^i$ . Therefore,  $M(1) = \{110, 111\}$ ,  $M(2) = \{110, 111\}$ , and  $M(3) = \{001, 111\}$ . Meanwhile,  $P_{110}^i$  has  $X_1$  and  $X_2$  as factors,  $P_{001}^i$  has  $X_3$  as a factor, and  $P_{111}^i$  has  $X_1$ ,  $X_2$ , and  $X_3$  as factors. Hence,  $N(110) = \{1, 2\}$ ,  $N(001) = \{3\}$ , and  $N(111) = \{1, 2, 3\}$ .

In order to see whether  $X_1$  is a PIS, by (17), we have

$$\frac{\sum_{l_{1,3} \in M(1)} P_{l_{1,3}}^i}{X_1(i)} = X_2(i) - X_2(i) X_3(i)$$

$$= \begin{cases} 0, & X_2(i) = 0, X_3(i) = 0, \\ 1, & X_2(i) = 1, X_3(i) = 0, \\ 0, & X_2(i) = 0, X_3(i) = 1, \\ 0, & X_2(i) = 1, X_3(i) = 1. \end{cases} \quad (18)$$

$\forall X_2(i), X_3(i), (\sum_{l_{1,3} \in M(1)} P_{l_{1,3}}^i) / X_1(i) \geq 0$ . Thus, (15) is satisfied, and  $X_1$  is a PIS. Similarly,  $X_2$  and  $X_3$  are also PISs.  $S_{\text{pis}}$  for the stochastic circuit in Example 5 is  $\{1, 2, 3\}$ .

### 3.2. Mathematical Model Based on MCAS and BS

**Definition 6.** If a bit stream  $X$  has the following properties:

$$X(i) = \begin{cases} 1, & 1 \leq i \leq nE_X, \\ 0, & \text{else,} \end{cases} \quad (19)$$

where  $n$  is the length of  $X$ , it is called an MCAS.

For instance, an MCAS with  $n = 8$  and  $E_X = 3/8$  is as follows: 1, 1, 1, 0, 0, 0, 0, 0.

Consider the combinational stochastic circuit with the  $k$  input streams  $X_1, X_2, \dots, X_k$  and an output stream  $Z$ . In order to indicate the sequence type for each input stream, we introduce the state vector  $\mathbf{V} = (V_1, V_2, \dots, V_k)$ , where

$$V_r = \begin{cases} \text{MCAS,} & \text{if } X_r \text{ is an MCAS,} \\ \text{BS,} & \text{if } X_r \text{ is a BS.} \end{cases} \quad (20)$$

There are totally  $2^k$  possible states for  $\mathbf{V}$ . For instance, if  $k = 2$ , the four possible states are follows: (MCAS, MCAS), (MCAS, BS), (BS, BS), and (BS, MCAS). The expectation and variance of output stream  $Z$ , when state vector is  $\mathbf{V}$ , are denoted by  $E(\mathbf{V})$  and  $\text{Var}(\mathbf{V})$ , respectively.

As discussed in Section 2.3, in order to guarantee that the arithmetic computed by the combinational stochastic circuit, when state vector is  $\mathbf{V}$ , is the same as that in the case of BI,  $\mathbf{V}$  should satisfy

$$E(\mathbf{V}) = E(\text{BI}). \quad (21)$$

On the premise of (21), if the state vector  $\mathbf{V}$  satisfies

$$\text{Var}(\mathbf{V}) \leq \text{Var}(\text{BI}), \quad (22)$$

then the random error is not greater than that in the case of BI. Thus, the state vector  $\mathbf{V}$  satisfying (21) and (22) represents a random error reduction scheme for the combinational stochastic circuit.

In order to describe the properties of state vector  $\mathbf{V}$ , we define the following notations. Let  $S_m$  denote the set of subscripts of input streams that are MCASs; that is,  $S_m = \{r : V_r = \text{MCAS}\}$ . Let  $S_{l_{1,k}}^i$  denote the set of subscripts of input streams that are MCASs and factors of  $P_{l_{1,k}}^i$ ; that is,  $S_{l_{1,k}}^i = \{r : V_r = \text{MCAS}, r \in N(l_{1,k})\}$ . Let  $S_{pm}$  denote the set of subscripts of product terms, of which at least one factor is an MCAS; that is,  $S_{pm} = \{l_{1,k} : N(l_{1,k}) \cap S_m \neq \emptyset, l_{1,k} \in S_p\}$ . Let  $S_{pb}$  denote the set of subscripts of product terms whose factors are all BSs; that is,  $S_{pb} = \{l_{1,k} : N(l_{1,k}) \cap S_m = \emptyset, l_{1,k} \in S_p\}$ . For instance, in Example 5, assuming that the state vector  $\mathbf{V} = (\text{MCAS}, \text{BS}, \text{BS})$ , then  $S_m = \{1\}$ ,  $S_{110} = \{1\}$ ,  $S_{001} = \emptyset$ ,  $S_{111} = \{1\}$ ,  $S_{pm} = \{110, 111\}$ , and  $S_{pb} = \{001\}$ . We also denote by  $|S|$  the cardinal of the set  $S$ . Using the above notations, we have the following lemmas and theorems.

**Lemma 7.** For any combinational stochastic circuit, if the state vector  $\mathbf{V}$  satisfies

$$|S_{l_{1,k}}^i| \leq 1, \quad \forall l_{1,k} \in S_p, \quad (23)$$

then one has that  $E(\mathbf{V}) = E(\text{BI})$ .

*Proof.* See Appendix A.  $\square$



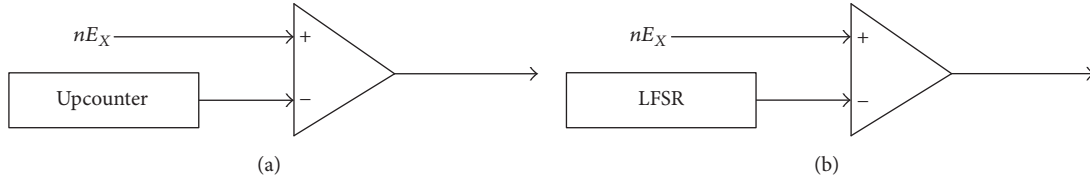


FIGURE 2: Hardware structure of the input stream generators: (a) MCAS generator; (b) BS generator.

**Lemma 8.** *Two input streams  $X_1$  and  $X_2$ , which are both MCASs, have the following properties:*

$$\sum_{i=1}^n \sum_{j \neq i} E(X_1(i)) = n(n-1)E_{X_1}, \quad (24)$$

$$\sum_{i=1}^n \sum_{j \neq i} E(X_2(j)) = n(n-1)E_{X_2}, \quad (25)$$

$$\sum_{i=1}^n \sum_{j \neq i} E(X_1(i))E(X_2(j)) \leq n(n-1)E_{X_1}E_{X_2}. \quad (26)$$

*Proof.* See Appendix B.  $\square$

**Theorem 9.** *For any combinational stochastic circuit, if the state vector  $\mathbf{V}$  satisfies*

$$|S_m| = 1, \quad (27)$$

*then one has  $E(\mathbf{V}) = E(BI)$ , and  $\text{Var}(\mathbf{V}) \leq \text{Var}(BI)$ .*

*Proof.* See Appendix C. Lemmas 7 and 8 will be used in the proof.  $\square$

**Theorem 10.** *For any combinational stochastic circuit, if the state vector  $\mathbf{V}$  satisfies*

$$S_m \subseteq S_{pis}, \quad (28)$$

$$|S_{l_{1,k}}| \leq 1, \quad \forall l_{1,k} \in S_p, \quad (29)$$

*then one has  $E(\mathbf{V}) = E(BI)$ , and  $\text{Var}(\mathbf{V}) \leq \text{Var}(BI)$ .*

*Proof.* See Appendix D. Lemmas 7 and 8 will be used in the proof.  $\square$

Theorems 9 and 10 give two different sufficient conditions for the state vector satisfying (21) and (22), respectively. On the basis of the two theorems, we will propose the random error reduction scheme for any combinational stochastic circuit in the next subsection.

**3.3. Random Error Reduction Scheme Based on MCAS and BS.** In this subsection, we illustrate how to get the state vector  $\mathbf{V}$  which represents the random error reduction scheme based on MCAS and BS for any combinational stochastic circuit. Specific steps are stated in Algorithm 1.

Now, we prove that the state vector  $\mathbf{V}$  obtained from Algorithm 1 satisfies (21) and (22). Consider the following

two cases: (I)  $S_{pis}$  is a null set; (II) otherwise. In case (I), the valid steps are (1), (2), (3), and (5). From criterion (a) in step (3), (27) is satisfied. According to Theorem 9, the state vector  $\mathbf{V}$  obtained from these steps satisfies (21) and (22). In case (II), the valid steps are (1), (2), (4), and (5). From criterion (a) and (b) in step (4), (28) and (29) are satisfied. According to Theorem 10, the state vector  $\mathbf{V}$  obtained from these steps also satisfies (21) and (22). Therefore, in both cases, the state vector  $\mathbf{V}$  obtained from the corresponding steps satisfies (21) and (22).

**3.4. Hardware Cost and Power Consumption.** In conventional stochastic computation, the input streams for stochastic circuits are all BSs. However, in the proposed random error reduction scheme, some input streams are specified as MCASs. Different input streams are generated by different generators. The hardware structures of the MCAS generator and the BS generator are shown in Figure 2. The MCAS generator consists of an  $n$ -bit comparator and an  $n$ -bit upcounter, while the BS generator consists of an  $n$ -bit comparator and an  $n$ -bit Linear Feedback Shift Register (LFSR), where  $2^n$  is the length of the input stream to be generated. In stochastic computation system, all the MCAS generators can share the same  $n$ -bit upcounter. By contrast, each BS generator requires a different  $n$ -bit LFSR. Thus, the total hardware cost of MCAS generators is usually lower than that of BS generators. That is to say, the proposed random error reduction scheme does not require any additional hardware cost.

Moreover, due to a lower switching activity, MCAS generator will have less dynamic power consumption than BS generator, which is another inherent advantage of our random error reduction scheme.

## 4. Case Study

For better understanding of the random error reduction scheme based on MCAS and BS, we take several typical stochastic circuits as case studies. We assume that all the input streams are uncorrelated.

**4.1. Case Study 1: AND Gate.** In this case study, we illustrate how to get the random error reduction scheme when  $S_{pis}$  is not a null set. An AND gate in stochastic logic implements a binary operator multiplication as  $z = x_1x_2$ . We obtain the state vector  $\mathbf{V}$  satisfying (21) and (22) from the steps given in Algorithm 1.

- (1) The relationship between input streams and output stream can be written as  $Z(i) = X_1(i)X_2(i)$ .

**Requirie:** The combinational stochastic circuit implementing arithmetic;

**Ensure:** The state vector  $\mathbf{V}$  representing the random error reduction scheme;

- (1) Express the relationship between input streams and output stream in the form of (14).
- (2) Obtain  $S_{\text{pis}}$ . If  $S_{\text{pis}}$  is a null set, go to step (3), otherwise, skip step (3) and go to step (4).
- (3) If  $S_{\text{pis}}$  is a null set, we determine which input streams are specified as MCASs and obtain  $S_m$  by the following criteria: (a) Eq. (27) should be satisfied, i.e., only one input stream is specified as MCAS; (b) on the premise of (a), let  $|S_{pm}|$  be maximal. Then, we skip step (4) and go to step (5).
- (4) If  $S_{\text{pis}}$  is not a null set, we determine which input streams are specified as MCASs and obtain  $S_m$  by the following criteria: (a) Eq. (28) should be satisfied, i.e., only PISs are likely to be specified as MCASs; (b) Eq. (29) should be satisfied, i.e., for each product term, at most one factor is specified as MCAS; (c) on the premise of (a) and (b), let  $|S_{pm}|$  be maximal. Then, we go to step (5).
- (5) The input streams not belonging to  $S_m$  are specified as BSs. From above, we determine each input stream should be specified as MCAS or BS and obtain the state vector  $\mathbf{V}$ .

ALGORITHM 1: Random error reduction scheme based on MCAS and BS.

TABLE 1: Candidates information in case study 1.

$S_m$	$S_{11}$	Valid	$S_{pm}$	$ S_{pm} $
{1}	{1}	Yes	{11}	1
{2}	{2}	Yes	{11}	1
{1,2}	{1,2}	No	Not concerned	Not concerned

(2)  $\forall X_2(i), (\sum_{l_{1,2} \in M(1)} P_{l_{1,2}}^i)/X_1(i) = X_2(i) \geq 0$ , and,  $\forall X_1(i), (\sum_{l_{1,2} \in M(1)} P_{l_{1,2}}^i)/X_2(i) = X_1(i) \geq 0$ . Thus, by Definition 4,  $X_1$  and  $X_2$  are PISs, and  $S_{\text{pis}} = \{1, 2\}$ . Since  $S_{\text{pis}}$  is not a null set, skip step (3) and go to step (4).

(3) Since  $S_{\text{pis}}$  is not a null set, this step is skipped.

(4) Since  $X_1$  and  $X_2$  are PISs, by criterion (a), both of them are likely to be specified as MCASs. Hence, as shown in the first column of Table 1,  $S_m$  is chosen from the following candidates: {1}, {2}, and {1, 2}. For each candidate of  $S_m$ , we get all the corresponding  $S_{l_{1,2}}$  and see whether criterion (b) is satisfied. If it is, the candidate is valid. For example, if  $S_m = \{1\}$ ,  $S_{11} = \{1\}$ . Moreover, there is only one product term  $P_{11}^i$ , and  $S_p = \{11\}$ . Then,  $\forall l_{1,2} \in S_p, |S_{l_{1,2}}| \leq 1$ , which is equivalent to (29). Criterion (b) is satisfied and the set {1} is a valid candidate. Similarly, the set {2} is also a valid candidate. However, if  $S_m = \{1, 2\}$ ,  $S_{11} = \{1, 2\}$ , which is opposed to (29). Thus, criterion (b) is not satisfied and the set {1, 2} is not a valid candidate. For each valid candidate, we get the corresponding  $|S_{pm}|$ . As shown in the last column of Table 1, when  $S_m = \{1\}$  or {2},  $|S_{pm}|$  is maximal, and criterion (c) is satisfied. From above, the set  $S_m$  satisfying all the criteria in step (4) can be written as  $S_m = \{1\}$  or {2}. Without loss of generality, we set  $S_m = \{1\}$ , and  $X_1$  is specified as an MCAS. Then, go to step (5).

(5)  $X_2$  is specified as a BS. Thus, the state vector  $\mathbf{V} =$  (MCAS, BS).

**4.2. Case Study 2: XOR Gate.** In this case study, we illustrate how to get random error reduction scheme when  $S_{\text{pis}}$  is a null set. An XOR gate in stochastic logic implements the arithmetic as follows:  $z = x_1 + x_2 - 2x_1x_2$ . We obtain the state vector  $\mathbf{V}$  satisfying (21) and (22) from the steps given in Algorithm 1.

(1) The relationship between input streams and output stream can be written in the form of (14) as  $Z(i) = X_1(i) + X_2(i) - 2X_1(i)X_2(i)$ .

(2) If  $X_2(i) = 1$ ,  $(\sum_{l_{1,2} \in M(1)} P_{l_{1,2}}^i)/X_1(i) = -1$ , and if  $X_1(i) = 1$ ,  $(\sum_{l_{1,2} \in M(1)} P_{l_{1,2}}^i)/X_2(i) = -1$ . Thus, by Definition 4,  $X_1$  and  $X_2$  are not PISs, and  $S_{\text{pis}}$  is a null set. Then, go to step (3).

(3) By criterion (a), only one of  $X_1$  and  $X_2$  can be specified as an MCAS. Hence, as shown in the first column of Table 2,  $S_m$  is chosen from the following candidates: {1} and {2}. For each candidate, we get the corresponding  $|S_{pm}|$ . As shown in the last column of Table 2, when  $S_m = \{1\}$  or {2},  $|S_{pm}|$  is maximal, and criterion (b) is satisfied. From above, the set  $S_m$  satisfying all the criteria in step (4) can be written as  $S_m = \{1\}$  or {2}. Without loss of generality, we set  $S_m = \{1\}$ , and  $X_1$  is specified as an MCAS. Then, skip step (4) and go to step (5).

(4) Since  $S_{\text{pis}}$  is a null set, this step is skipped.

(5)  $X_2$  is specified as a BS. Thus, the state vector  $\mathbf{V} =$  (MCAS, BS).

TABLE 2: Candidates information in case study 2.

$S_m$	$S_{10}$	$S_{01}$	$S_{11}$	$S_{pm}$	$ S_{pm} $
{1}	{1}	Null	{1}	{10, 11}	2
{2}	Null	{2}	{2}	{01, 11}	2

TABLE 3: Candidates information in case study 3.

$S_m$	$S_{10000}$	$S_{10010}$	$S_{10001}$	$S_{10011}$	$S_{01010}$	$S_{01001}$	$S_{01011}$	$S_{00111}$	Valid	$ S_{pm} $
{1}	{1}	{1}	{1}	{1}	Null	Null	Null	Null	Yes	4
{2}	Null	Null	Null	Null	{2}	{2}	{2}	Null	Yes	3
{3}	Null	Null	Null	Null	Null	Null	Null	{3}	Yes	1
{2, 3}	Null	Null	Null	Null	{2}	{2}	{2}	{3}	Yes	4
{1, 2}	{1}	{1}	{1}	{1}	{2}	{2}	{2}	Null	Yes	7
{1, 3}	{1}	{1}	{1}	{1}	Null	Null	Null	{3}	Yes	5
{1, 2, 3}	{1}	{1}	{1}	{1}	{2}	{2}	{2}	{3}	Yes	8

**4.3. Case Study 3: Stochastic Logic Implementing Bernstein Polynomial.** In this case study, we illustrate how to get the random error reduction scheme for the stochastic logic implementing Bernstein polynomial with coefficients in the unit interval. A Bernstein polynomial of degree  $k$ , denoted by  $B_k(x)$ , is defined by  $B_k(x) = \sum_{i=0}^k b_i \binom{k}{i} x^i (1-x)^{k-i}$ , where each  $b_i$  is a constant coefficient. In [4], Qian et al. propose the stochastic circuit for the Bernstein polynomial with coefficients in the unit interval, which is shown in Figure 3. For a Bernstein polynomial of degree  $k$ , there are  $2k + 1$  input streams:  $X_1, X_2, \dots, X_{2k+1}$ , with the following properties:  $E_{X_1} = b_0, E_{X_2} = b_1, \dots, E_{X_{k+1}} = b_k$ , and  $E_{X_{k+2}} = E_{X_{k+3}} = \dots = E_{X_{2k+1}} = x$ . For simplicity, we will take the Bernstein polynomial of degree 2 as an example, where

$$\begin{aligned}
Z(i) &= X_1(i)(1 - X_4(i))(1 - X_5(i)) \\
&\quad + X_2(i)X_4(i)(1 - X_5(i)) \\
&\quad + X_2(i)(1 - X_4(i))X_5(i) \\
&\quad + X_3(i)X_4(i)X_5(i).
\end{aligned} \tag{30}$$

We obtain the state vector  $\mathbf{V}$  satisfying (21) and (22) from the steps given in Algorithm 1.

- (1) The relationship between input streams and output stream can be rewritten in the form of (14) as

$$\begin{aligned}
Z(i) &= X_1(i) - X_1(i)X_4(i) - X_1(i)X_5(i) \\
&\quad + X_1(i)X_4(i)X_5(i) + X_2(i)X_4(i) \\
&\quad + X_2(i)X_5(i) - 2X_2(i)X_4(i)X_5(i) \\
&\quad + X_3(i)X_4(i)X_5(i).
\end{aligned} \tag{31}$$

- (2)  $\forall X_2(i), X_3(i), X_4(i), X_5(i), (\sum_{l_{1,5} \in M(1)} P_{l_{1,5}}^i) / X_1(i) = 1 - X_4(i) - X_5(i) + X_4(i)X_5(i) \geq 0$ . Thus, by Definition 4,  $X_1$  is a PIS. Similarly,  $X_2$  and  $X_3$  are also PISs, and  $S_{\text{pis}} = \{1, 2, 3\}$ . Since  $S_{\text{pis}}$  is not a null set, skip step (3) and go to step (4).

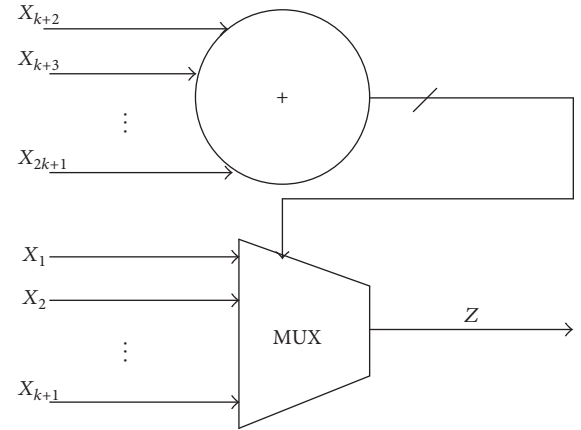


FIGURE 3: Stochastic logic implementing Bernstein polynomial.

- (3) Since  $S_{\text{pis}}$  is not a null set, this step is skipped.
- (4) Since  $X_1, X_2$ , and  $X_3$  are PISs, by criterion (a), all of them are likely to be specified as MCASs. Hence, as shown in the first column of Table 3,  $S_m$  is chosen from the following candidates: {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, and {1, 2, 3}. For each candidate of  $S_m$ , we get all the corresponding  $S_{l_{1,5}}$  and see whether criterion (b) is satisfied. If it is, the candidate is valid. For example, if  $S_m = \{1\}$ , then  $S_{10000} = \{1\}$ ,  $S_{10010} = \{1\}$ ,  $S_{10001} = \{1\}$ ,  $S_{10011} = \{1\}$ ,  $S_{01010} = \emptyset$ ,  $S_{01001} = \emptyset$ ,  $S_{01011} = \emptyset$ , and  $S_{00111} = \emptyset$ . Moreover, there are eight product terms, and  $S_p = \{10000, 10010, 10001, 10011, 01010, 01001, 01011, 00111\}$ . Hence,  $\forall l_{1,5} \in S_p, |S_{l_{1,5}}| \leq 1$ , which is equivalent to (29). Criterion (b) is satisfied and the set {1} is a valid candidate. Similarly, the other sets in the first column of Table 3 are also valid candidates. For each valid candidate, we get the corresponding  $|S_{pm}|$ . As shown in the last column of Table 3, when  $S_m = \{1, 2, 3\}$ ,  $|S_{pm}|$  is maximal, and criterion (c) is satisfied. From



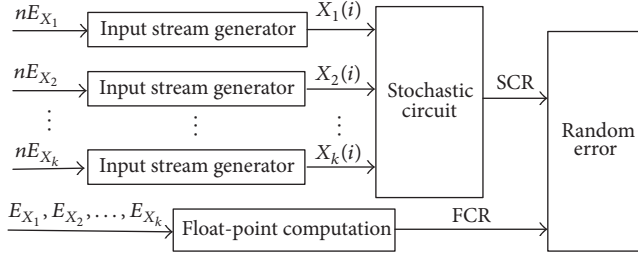


FIGURE 4: Simulation platform for evaluating random error.

above, the set  $S_m$  satisfying all the criteria in step (4) can be written as  $S_m = \{1, 2, 3\}$ . Then, go to step (5).

- (5)  $X_4$  and  $X_5$  are specified as BSs. Thus, the state vector  $\mathbf{V} = (\text{MCAS}, \text{MCAS}, \text{MCAS}, \text{BS}, \text{BS})$ .

Similarly, we can get the state vector  $\mathbf{V}$  satisfying (21) and (22) for the stochastic circuit implementing Bernstein polynomial of degree  $n$ , which can be written as  $\mathbf{V} = (\text{MCAS}, \text{MCAS}, \text{BS}, \text{BS})$ .

## 5. Simulation Results and Comparison

The simulation platform for evaluating the random error in stochastic circuits is shown in Figure 4. Assume that the stochastic circuit is with  $k$  input streams  $X_1, X_2, \dots, X_k$ , which represent  $k$  variables for the corresponding arithmetic. We randomly select 5000 sample points of  $E_{X_1}, E_{X_2}, \dots, E_{X_k}$  from  $[0, 1]$ . These sample points are used in both stochastic computation and float-point computation. The random error can be measured by the average of the absolute difference between the stochastic computation results (SCR) and the floating-point computation results (FCR) on these sample points. In order to simulate the generic and common circumstance, we used the same LFSR for all the input stream generators and just change the seed value. The size of LFSR equals the square root of length of input streams. For the sake of illustrating that the random error reduction scheme is valid for any input stream length, the length of input streams is chosen to be  $2^m$ , where  $m = 5, 6, \dots, 12$ . We compare the random error based on MCAS and BS with that based on other sequences on the following benchmarks.

- (1) AND gate: the random error reduction scheme based on MCAS and BS for the stochastic logic AND gate is as follows:  $X_1$  is specified as an MCAS, and  $X_2$  is specified as a BS. For comparison, as shown in Figure 5, we give the simulation results in the following cases: (a)  $X_1$  is an MCAS, and  $X_2$  is a BS; (b)  $X_1$  and  $X_2$  are both BSs [4]; (c)  $X_1$  and  $X_2$  are both FRSs [14]; (d)  $X_1$  and  $X_2$  are DBSs as illustrated in [3].
- (2) XOR gate: the random error reduction scheme based on MCAS and BS for the stochastic logic XOR gate is as follows:  $X_1$  is specified as an MCAS, and  $X_2$  is specified as a BS. For comparison, as shown in Figure 6, we give the simulation results in the

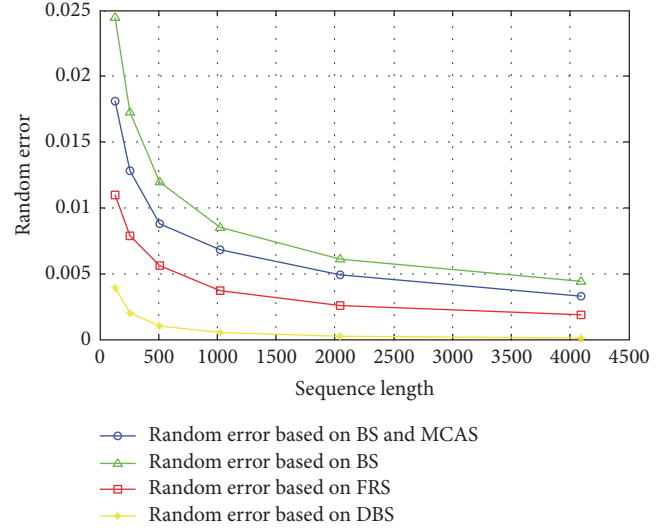


FIGURE 5: Simulation results for AND gate.

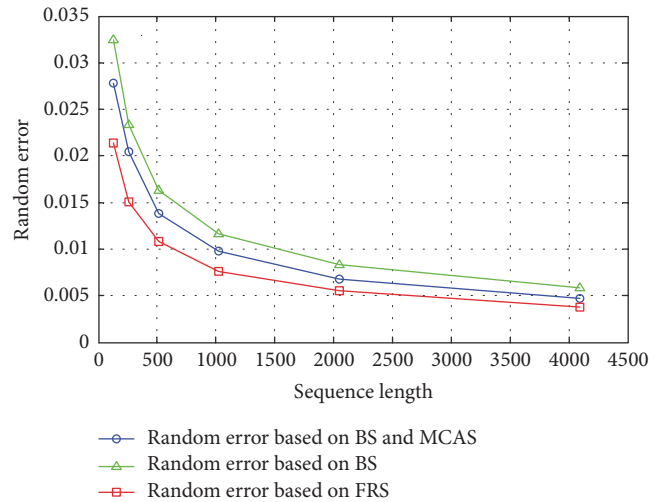
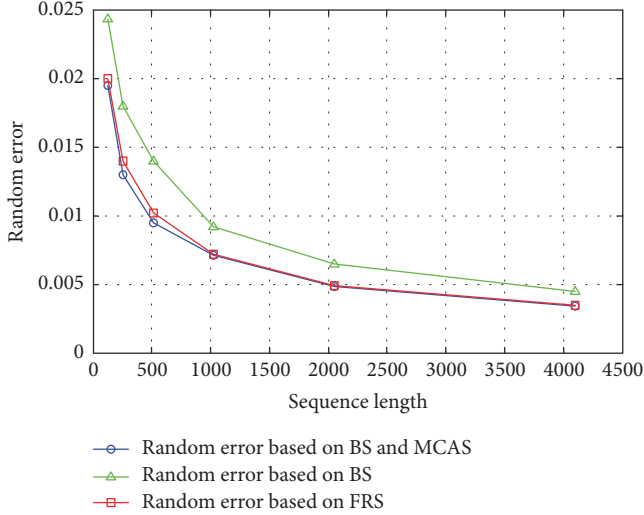
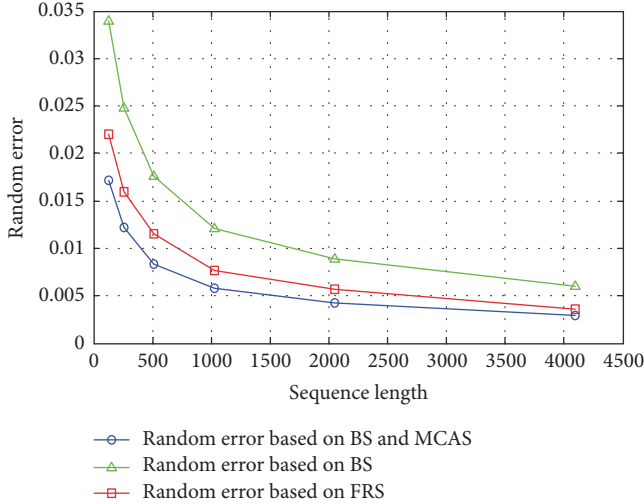


FIGURE 6: Simulation results for XOR gate.

following cases: (a)  $X_1$  is an MCAS, and  $X_2$  is a BS; (b)  $X_1$  and  $X_2$  are both BSs [4]; (c)  $X_1$  and  $X_2$  are both FRSs [14].

- (3) Stochastic logic implementing Bernstein polynomial: two Bernstein polynomials are referred to in [4]. The one used for approximating gamma correction function is of degree 6 and with the following coefficients:  $b_0 = 0.0955, b_1 = 0.7207, b_2 = 0.3476, b_3 = 0.9988, b_4 = 0.7017, b_5 = 0.9695$ , and  $b_6 = 0.9939$ . The other used for synthesizing polynomials is of degree 3 and with the coefficients as follows:  $b_0 = 0.2500, b_1 = 0.6250, b_2 = 0.3750$ , and  $b_3 = 0.7500$ . We denote the stochastic circuit implementing the former Bernstein polynomial by  $B_1$  and the later by  $B_2$ . As illustrated in case study 3, the random error reduction scheme based on MCAS and BS for  $B_1$  is as follows:  $X_1, X_2, \dots, X_7$  are specified as MCASs, and  $X_8, X_9, \dots, X_{13}$  are specified as BSs. Meanwhile, that

FIGURE 7: Simulation results for  $B_1$ .FIGURE 8: Simulation results for  $B_2$ .

for  $B_2$  is as follows:  $X_1, X_2, \dots, X_5$  are specified as MCASs, and  $X_6, X_7, \dots, X_9$  are specified as BSs. As shown in Figure 7, we give the simulation results for  $B_1$  in three cases: (a)  $X_1, X_2, \dots, X_7$  are MCASs, and  $X_8, X_9, \dots, X_{13}$  are BSs; (b) all the input streams are BSs; (c) all the input streams are FRSs. Similarly, as shown in Figure 8, we also give out the simulation results for  $B_2$  in three cases: (a)  $X_1, X_2, \dots, X_5$  are MCASs, and  $X_6, X_7, \dots, X_9$  are BSs; (b) all the input streams are BSs [4]; (c) all the input streams are FRSs [14].

From the simulation results for the above benchmarks, we have the following conclusions.

- (i) For all the benchmarks, the random error based on MCAS and BS is smaller than that based on BS with any sequence length. Hence, the proposed

random error reduction scheme is effective for those stochastic circuits.

- (ii) In benchmark 1, DBSSs have the best performance. However, they are only suitable for this specific stochastic circuit. By comparison, the proposed MCASs and BSs can be used in any combination stochastic circuit.
- (iii) FRSs have better performance than MCASs and BSs in benchmarks 1 and 2, and vice versa in benchmark 3. Considering the difficulties in hardware generation for FRSs, the proposed MCASs and BSs, easily generated by hardware, have much wider applications.
- (iv) For all the benchmarks, the random error reduction scheme based on MCAS and BS is valid for any input stream length. Hence, the proposed random error reduction scheme has no length constraints.

## 6. Conclusion

In this paper, we propose a general random error analysis method based on AS. According to the analysis results, we find it feasible to reduce random error for stochastic computation by using proper ASs as input streams. Then, we present the random error reduction scheme based on MCAS and BS, which has the advantages of wide applications, easy hardware implementation, and flexible length. Both the theoretical analysis and simulation results confirm the effectiveness of the proposed scheme. In the future work, we will further study the features of ASs and apply them into more complex stochastic circuits.

## Appendix

### A. Proof of Lemma 7

*Proof.* By (23), for any product term, at most one factor is an MCAS. Let  $X^{l_{1,k}}$  denote the input stream that is an MCAS and a factor of  $P_{l_{1,k}}^i$ . Then, we have

$$E(P_{l_{1,k}}^i) = A_{l_{1,k}} \prod_{v \in N(l_{1,k})} E(X_v(i))$$

$$= \begin{cases} C_{l_{1,k}}, & l_{1,k} \in S_{pb}, \\ \frac{C_{l_{1,k}} E(X^{l_{1,k}}(i))}{E_{X^{l_{1,k}}}}, & l_{1,k} \in S_{pm}, \end{cases} \quad (\text{A.1})$$

where  $C_{l_{1,k}} = A_{l_{1,k}} \prod_{v \in N(l_{1,k})} E_{X_v}$ . The expectation of output stream can be written as

$$E_Z = \frac{1}{n} \sum_{i=1}^n E(Z(i)) = \frac{1}{n} \sum_{i=1}^n \sum_{l_{1,k} \in S_p} E(P_{l_{1,k}}^i). \quad (\text{A.2})$$

If all the input streams are BSs,  $S_p = S_{pb}$ . Substituting (A.1) into (A.2) we have

$$E(\text{BI}) = \frac{1}{n} \sum_{i=1}^n \sum_{l_{1,k} \in S_{pb}} E(P_{l_{1,k}}^i) = \sum_{l_{1,k} \in S_p} C_{l_{1,k}}. \quad (\text{A.3})$$

If the state vector  $V$  satisfies (23),  $S_p = S_{pm} \cup S_{pb}$ . Substituting (A.1) into (A.2) we have

$$\begin{aligned} E(\mathbf{V}) &= \frac{1}{n} \sum_{i=1}^n \sum_{l_{1,k} \in S_{pb}} E(P_{l_{1,k}}^i) + \frac{1}{n} \sum_{i=1}^n \sum_{l_{1,k} \in S_{pm}} E(P_{l_{1,k}}^i) \\ &= \sum_{l_{1,k} \in S_{pb}} C_{l_{1,k}} \\ &\quad + \sum_{l_{1,k} \in S_{pm}} C_{l_{1,k}} \left( \frac{1}{n} \frac{\sum_{i=1}^n E(X^{l_{1,k}}(i))}{E_{X^{l_{1,k}}}} \right) \\ &= \sum_{l_{1,k} \in S_{pb}} C_{l_{1,k}} + \sum_{l_{1,k} \in S_{pm}} C_{l_{1,k}} = \sum_{l_{1,k} \in S_p} C_{l_{1,k}}. \end{aligned} \quad (\text{A.4})$$

Combining (A.3) and (A.4),  $E(\mathbf{V}) = E(\text{BI})$ , and Lemma 7 is proved.  $\square$

## B. Proof of Lemma 8

*Proof.*

$$\sum_{i=1}^n \sum_{j=1}^n E(X_1(i)) = \sum_{j=1}^n \left( \sum_{i=1}^n E(X_1(i)) \right) = n^2 E_{X_1}. \quad (\text{B.1})$$

$$\sum_{i=1}^n \sum_{j=i}^n E(X_1(i)) = \sum_{j=i}^n \left( \sum_{i=1}^n E(X_1(i)) \right) = n E_{X_1}. \quad (\text{B.2})$$

Combining (B.1) and (B.2), we have

$$\sum_{i=1}^n \sum_{j \neq i}^n E(X_1(i)) = n(n-1) E_{X_1}. \quad (\text{B.3})$$

Similarly,

$$\sum_{i=1}^n \sum_{j \neq i}^n E(X_2(j)) = n(n-1) E_{X_2}. \quad (\text{B.4})$$

Since  $X_1$  and  $X_2$  are MCASs,

$$E(X_1(i)) E(X_2(i)) = \begin{cases} 1, & i \leq \min(nE_{X_1}, nE_{X_2}), \\ 0, & \text{otherwise.} \end{cases} \quad (\text{B.5})$$

Meanwhile,

$$\begin{aligned} &\sum_{i=1}^n \sum_{j=i}^n E(X_1(i)) E(X_2(j)) \\ &= \sum_{i=1}^n E(X_1(i)) E(X_2(i)). \end{aligned} \quad (\text{B.6})$$

Substituting (B.5) into (B.6), we have

$$\sum_{i=1}^n \sum_{j=i}^n E(X_1(i)) E(X_2(j)) = n \min(E_{X_1}, E_{X_2}). \quad (\text{B.7})$$

Moreover,

$$\begin{aligned} &\sum_{i=1}^n \sum_{j=1}^n E(X_1(i)) E(X_2(j)) \\ &= \sum_{i=1}^n E(X_1(i)) \sum_{j=1}^n E(X_2(j)) = n^2 E_{X_1} E_{X_2}. \end{aligned} \quad (\text{B.8})$$

By (B.7) and (B.8), we have

$$\begin{aligned} &\sum_{i=1}^n \sum_{j \neq i}^n E(X_1(i)) E(X_2(j)) \\ &= \sum_{i=1}^n \sum_{j=1}^n E(X_1(i)) E(X_2(j)) \\ &\quad - \sum_{i=1}^n \sum_{j=i}^n E(X_1(i)) E(X_2(j)) \\ &= n^2 E_{X_1} E_{X_2} - n \min(E_{X_1}, E_{X_2}) \\ &\leq (n^2 - n) E_{X_1} E_{X_2}. \end{aligned} \quad (\text{B.9})$$

Combining (B.3), (B.4), and (B.9), Lemma 8 is proved.  $\square$

## C. Proof of Theorem 9

*Proof.* From (27), only one input stream is specified as an MCAS when the state vector is  $\mathbf{V}$ . Thus, (23) is obviously satisfied. According to Lemma 7, we have that  $E(\mathbf{V}) = E(\text{BI})$ . Then we will prove that  $\mathbf{V}$  also satisfies (22).

$$\begin{aligned} &E(P_{l_{1,k}}^i P_{h_{1,k}}^j) \\ &= A_{l_{1,k}} A_{h_{1,k}} \prod_{v \in N(l_{1,k})} E(X_v(i)) \prod_{v \in N(h_{1,k})} E(X_u(i)). \end{aligned} \quad (\text{C.1})$$

Let  $X_f$  denote the input streams that are an MCAS. Then, (C.1) can be rewritten as

$$\begin{aligned} &E(P_{l_{1,k}}^i P_{h_{1,k}}^j) \\ &= \begin{cases} C_{l_{1,k}}^{h_{1,k}}, & l_{1,k} \notin M(f), h_{1,k} \notin M(f), \\ \frac{C_{l_{1,k}}^{h_{1,k}} E(X_f(i))}{E_{X_f}}, & l_{1,k} \in M(f), h_{1,k} \notin M(f), \\ \frac{C_{l_{1,k}}^{h_{1,k}} E(X_f(i))}{E_{X_f}}, & l_{1,k} \notin M(f), h_{1,k} \in M(f), \\ \frac{C_{l_{1,k}}^{h_{1,k}} E(X_f(i)) E(X_f(j))}{(E_{X_f})^2}, & l_{1,k} \in M(f), h_{1,k} \in M(f), \end{cases} \end{aligned} \quad (\text{C.2})$$

where  $C_{l_{1,k}}^{h_{1,k}} = A_{l_{1,k}} A_{h_{1,k}} \prod_{v \in N(l_{1,k})} E_{X_v} \prod_{u \in N(h_{1,k})} E_{X_u} = C_{l_{1,k}} C_{h_{1,k}}$ . From (27), we have

$$\begin{aligned} & \sum_{i=1}^n \sum_{j \neq i} E(Z(i) Z(j)) \\ &= \sum_{l_{1,k} \in S_p} \sum_{h_{1,k} \in S_p} \sum_{i=1}^n \sum_{j \neq i} E(P_{l_{1,k}}^i P_{h_{1,k}}^j). \end{aligned} \quad (C.3)$$

For illustration purposes, we make the following notations:

$$\begin{aligned} F_1 &= \sum_{l_{1,k} \notin M(f)} \sum_{h_{1,k} \notin M(f)} \sum_{i=1}^n \sum_{j \neq i} E(P_{l_{1,k}}^i P_{h_{1,k}}^j), \\ F_2 &= \sum_{l_{1,k} \in M(f)} \sum_{h_{1,k} \notin M(f)} \sum_{i=1}^n \sum_{j \neq i} E(P_{l_{1,k}}^i P_{h_{1,k}}^j), \\ F_3 &= \sum_{l_{1,k} \notin M(f)} \sum_{h_{1,k} \in M(f)} \sum_{i=1}^n \sum_{j \neq i} E(P_{l_{1,k}}^i P_{h_{1,k}}^j), \\ F_4 &= \sum_{l_{1,k} \in M(f)} \sum_{h_{1,k} \in M(f)} \sum_{i=1}^n \sum_{j \neq i} E(P_{l_{1,k}}^i P_{h_{1,k}}^j). \end{aligned} \quad (C.4)$$

Substituting (C.2) into  $F_1$ , we have

$$F_1 = n(n-1) \sum_{l_{1,k} \notin M(f)} \sum_{h_{1,k} \notin M(f)} C_{l_{1,k}}^{h_{1,k}}. \quad (C.5)$$

Substituting (24) and (C.2) into  $F_2$ ,

$$\begin{aligned} F_2 &= \frac{\sum_{l_{1,k} \in M(f)} \sum_{h_{1,k} \notin M(f)} C_{l_{1,k}}^{h_{1,k}} \sum_{i=1}^n \sum_{j \neq i} E(X_f(i))}{E_{X_f}} \\ &= n(n-1) \sum_{l_{1,k} \in M(f)} \sum_{h_{1,k} \notin M(f)} C_{l_{1,k}}^{h_{1,k}}. \end{aligned} \quad (C.6)$$

Similarly, substituting (25) and (C.2) into  $F_3$ ,

$$F_3 = n(n-1) \sum_{l_{1,k} \notin M(f)} \sum_{h_{1,k} \in M(f)} C_{l_{1,k}}^{h_{1,k}}. \quad (C.7)$$

Moreover, substituting (C.2) into  $F_4$ ,

$$\begin{aligned} F_4 &= \frac{\sum_{l_{1,k} \in M(f)} \sum_{h_{1,k} \in M(f)} C_{l_{1,k}}^{h_{1,k}} \sum_{i=1}^n \sum_{j \neq i} E(X_f(i)) E(X_f(j))}{(E_{X_f})^2}, \end{aligned} \quad (C.8)$$

where  $\sum_{l_{1,k} \in M(f)} \sum_{h_{1,k} \in M(f)} C_{l_{1,k}}^{h_{1,k}} = (\sum_{l_{1,k} \in M(f)} C_{l_{1,k}})^2 \geq 0$ . By (26) and (C.8),

$$F_4 \leq n(n-1) \sum_{l_{1,k} \in M(f)} \sum_{h_{1,k} \in M(f)} C_{l_{1,k}}^{h_{1,k}}. \quad (C.9)$$

Combining (C.5), (C.6), (C.7), and (C.9),

$$\begin{aligned} \sum_{i=1}^n \sum_{j \neq i} E(Z(i) Z(j)) &\leq n(n-1) \sum_{l_{1,k} \in S_p} \sum_{h_{1,k} \in S_p} C_{l_{1,k}}^{h_{1,k}} \\ &= n(n-1) \left( \sum_{l_{1,k} \in S_p} C_{l_{1,k}} \right)^2. \end{aligned} \quad (C.10)$$

By (A.3) and (C.10),

$$\frac{1}{n^2} \sum_{i=1}^n \sum_{j \neq i} E(Z(i) Z(j)) \leq \frac{n-1}{n} E^2(\text{BI}). \quad (C.11)$$

By (12) and (C.11), we have

$$\text{Var}(\mathbf{V}) \leq \frac{E(\text{BI})(1 - E(\text{BI}))}{n} = \text{Var}(\text{BI}). \quad (C.12)$$

From above, Theorem 9 is proved.  $\square$

## D. Proof of Theorem 10

*Proof.* We can see that (29) is the same as (23). Thus, by Lemma 7, we have  $E(\mathbf{V}) = E(\text{BI})$ . Then we will prove that  $\mathbf{V}$  also satisfies (22). Let  $X^{l_{1,k}}$  denote the input stream that is an MCAS and a factor of  $P_{l_{1,k}}^i$ . Let  $X^{h_{1,k}}$  denote the input stream that is an MCAS and a factor of  $P_{h_{1,k}}^j$ . Then, (C.1) can be rewritten as

$$E(P_{l_{1,k}}^i P_{h_{1,k}}^j) = \begin{cases} C_{l_{1,k}}^{h_{1,k}}, & l_{1,k} \notin S_{pm}, h_{1,k} \notin S_{pm}, \\ \frac{C_{l_{1,k}}^{h_{1,k}} E(X^{l_{1,k}}(i))}{E_{X^{l_{1,k}}}}, & l_{1,k} \in S_{pm}, h_{1,k} \notin S_{pm}, \\ \frac{C_{l_{1,k}}^{h_{1,k}} E(X^{h_{1,k}}(j))}{E_{X^{h_{1,k}}}}, & l_{1,k} \notin S_{pm}, h_{1,k} \in S_{pm}, \\ \frac{C_{l_{1,k}}^{h_{1,k}} E(X^{l_{1,k}}(i)) E(X^{h_{1,k}}(j))}{(E_{X^{l_{1,k}}} E_{X^{h_{1,k}}})}, & l_{1,k} \in S_{pm}, h_{1,k} \in S_{pm}. \end{cases} \quad (D.1)$$

For illustration purposes, we make the following notations:

$$\begin{aligned} G_1 &= \sum_{l_{1,k} \notin S_{pm}} \sum_{h_{1,k} \notin S_{pm}} \sum_{i=1}^n \sum_{j \neq i} E(P_{l_{1,k}}^i P_{h_{1,k}}^j), \\ G_2 &= \sum_{l_{1,k} \in S_{pm}} \sum_{h_{1,k} \notin S_{pm}} \sum_{i=1}^n \sum_{j \neq i} E(P_{l_{1,k}}^i P_{h_{1,k}}^j), \\ G_3 &= \sum_{l_{1,k} \notin S_{pm}} \sum_{h_{1,k} \in S_{pm}} \sum_{i=1}^n \sum_{j \neq i} E(P_{l_{1,k}}^i P_{h_{1,k}}^j), \\ G_4 &= \sum_{l_{1,k} \in S_{pm}} \sum_{h_{1,k} \in S_{pm}} \sum_{i=1}^n \sum_{j \neq i} E(P_{l_{1,k}}^i P_{h_{1,k}}^j). \end{aligned} \quad (D.2)$$

Substituting (D.1) into  $G_1$ ,

$$G_1 = n(n-1) \sum_{l_{1,k} \notin S_{pm}} \sum_{h_{1,k} \notin S_{pm}} C_{l_{1,k}}^{h_{1,k}}. \quad (D.3)$$

Substituting (24) and (D.1) into  $G_2$ ,

$$\begin{aligned} G_2 &= \frac{\sum_{l_{1,k} \in S_{pm}} \sum_{h_{1,k} \notin S_{pm}} C_{l_{1,k}}^{h_{1,k}} \sum_{i=1}^n \sum_{j \neq i} E(X^{l_{1,k}}(i))}{E_{X^{l_{1,k}}}} \\ &= n(n-1) \sum_{l_{1,k} \in S_{pm}} \sum_{h_{1,k} \notin S_{pm}} C_{l_{1,k}}^{h_{1,k}}. \end{aligned} \quad (D.4)$$

Similarly, substituting (25) and (D.1) into  $G_3$ ,

$$G_3 = n(n-1) \sum_{l_{1,k} \notin S_{pm}} \sum_{h_{1,k} \in S_{pm}} C_{l_{1,k}}^{h_{1,k}}. \quad (D.5)$$

Moreover, substituting (D.1) into  $G_4$ ,

$$\begin{aligned} G_4 &= \frac{\sum_{l_{1,k} \in S_{pm}} \sum_{h_{1,k} \in S_{pm}} C_{l_{1,k}}^{h_{1,k}} \sum_{i=1}^n \sum_{j \neq i} E(X^{l_{1,k}}(i)) E(X^{h_{1,k}}(j))}{(E_{X^{l_{1,k}}} E_{X^{h_{1,k}}})} \\ &= \frac{\sum_{e \in S_m} \sum_{l_{1,k} \in M(e)} \sum_{f \in S_m} \sum_{h_{1,k} \in M(f)} C_{l_{1,k}}^{h_{1,k}} \sum_{i=1}^n \sum_{j \neq i} E(X_e(i)) E(X_f(j))}{(E_{X_e} E_{X_f})}. \end{aligned} \quad (D.6)$$

According to (28), all the input streams that are MCASs should be PISs. Thus,  $X_e$  and  $X_f$  are both PISs. By Definition 6,  $\forall X_1(i), X_2(i), \dots, X_{e-1}(i), X_{e+1}(i), \dots, X_k(i)$ ,  $(\sum_{l_{1,k} \in M(e)} P_{l_{1,k}}^i) / X_e(i) \geq 0$ . Thus, we have

$$E\left(\frac{\sum_{l_{1,k} \in M(e)} P_{l_{1,k}}^i}{X_e(i)}\right) = \frac{\sum_{l_{1,k} \in M(e)} C_{l_{1,k}}}{E_{X_e}} \geq 0. \quad (D.7)$$

Similarly,

$$\frac{\sum_{h_{1,k} \in M(f)} C_{h_{1,k}}}{E_{X_f}} \geq 0. \quad (D.8)$$

Combining (D.7) and (D.8),

$$\sum_{l_{1,k} \in M(e)} \sum_{h_{1,k} \in M(f)} C_{l_{1,k}}^{h_{1,k}} \geq 0. \quad (D.9)$$

By (26), (D.6), and (D.9),

$$\begin{aligned} G_4 &\leq n(n-1) \sum_{e \in S_m} \sum_{l_{1,k} \in M(e)} \sum_{f \in S_m} \sum_{h_{1,k} \in M(f)} C_{l_{1,k}}^{h_{1,k}} \\ &= n(n-1) \sum_{l_{1,k} \in S_{pm}} \sum_{h_{1,k} \in S_{pm}} C_{l_{1,k}}^{h_{1,k}}. \end{aligned} \quad (D.10)$$

Combining (D.3), (D.4), (D.5), and (D.10),

$$\begin{aligned} \sum_{i=1}^n \sum_{j \neq i} E(Z(i)Z(j)) &\leq n(n-1) \sum_{l_{1,k} \in S_p} \sum_{h_{1,k} \in S_p} C_{l_{1,k}}^{h_{1,k}} \\ &= n(n-1) \left( \sum_{l_{1,k} \in S_p} C_{l_{1,k}} \right)^2. \end{aligned} \quad (D.11)$$

By (A.3) and (D.11),

$$\frac{1}{n^2} \sum_{i=1}^n \sum_{j \neq i} E(Z(i)Z(j)) \leq \frac{n-1}{n} E^2(\text{BI}) \quad (D.12)$$

By (12) and (D.12), we have

$$\text{Var}(\mathbf{V}) \leq \frac{E(\text{BI})(1 - E(\text{BI}))}{n} = \text{Var}(\text{BI}). \quad (D.13)$$

From above, Theorem 10 is proved.  $\square$

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61371104) and the National High Technology Project of China (Grants nos. 2014AA01A707 and 2014AA01A070).

## References

- [1] B. R. Gaines, "Stochastic computing," in *Proceedings of the Spring Joint Computer Conference*, pp. 149–156, Atlantic City, NJ, USA, April 1967.
- [2] B. D. Brown and H. C. Card, "Stochastic neural computation I: computational elements," *IEEE Trans Comput*, vol. 50, no. 9, pp. 891–905, 2001.
- [3] D. Braendler, T. Hendtlass, and P. O'Donoghue, "Deterministic bit-stream digital neurons," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1514–1525, 2002.
- [4] W. 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, 2011.
- [5] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. D. Riedel, "Computation on stochastic bit streams digital image processing case studies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 3, pp. 449–462, 2014.
- [6] P. Li, D. J. Lilja, W. Qian, M. D. Riedel, and K. Bazargan, "Logical computation on stochastic bit streams with linear finite-state machines," *IEEE Transactions on Computers*, vol. 63, no. 6, pp. 1473–1485, 2014.



- [7] M. H. Najafi and M. E. Salehi, "A Fast fault-tolerant architecture for sauvola local image thresholding algorithm using stochastic computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 2, pp. 808–812, 2016.
- [8] S. Sharifi Tehrani, S. Mannor, and W. J. Gross, "Fully parallel stochastic LDPC decoders," *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5692–5703, 2008.
- [9] Q. T. Dong, M. Arzel, C. Jego, and W. J. Gross, "Stochastic decoding of turbo codes," *IEEE Transactions on Signal Processing*, vol. 58, no. 12, pp. 6421–6425, 2010.
- [10] A. Naderi, S. Mannor, M. Sawan, and W. J. Gross, "Delayed stochastic decoding of LDPC codes," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5617–5626, 2011.
- [11] J. Chen, J. Hu, and G. E. Sobelman, "Stochastic mimo detector based on the markov chain monte carlo algorithm," *IEEE Transactions on Signal Processing*, vol. 62, no. 6, pp. 1454–1463, 2014.
- [12] J. Chen, J. Hu, and G. E. Sobelman, "tochastic iterative mimo detection system: algorithm and hardware design," *IEEE Transactions on Circuits and Systems. I*, vol. 62, no. 4, pp. 1205–1214, 2015.
- [13] J. Han, H. Chen, J. Liang, P. Zhu, Z. Yang, and F. Lombardi, "A stochastic computational approach for accurate and efficient reliability evaluation," *IEEE Transactions on Computers*, vol. 63, no. 6, pp. 1336–1350, 2014.
- [14] A. Alaghi and J. P. Hayes, "Exploiting correlation in stochastic circuit design," in *Proceedings of the IEEE International Conference on Computer Design*, pp. 39–46, Asheville, NC, USA, October 2013.
- [15] A. Alaghi and J. P. Hayes, "Strauss: spectral transform use in stochastic circuit synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 11, pp. 1770–1783, 2015.
- [16] M. H. Najafi, S. Jamali-Zavareh et al., "Time-encoded values for highly efficient stochastic circuits," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 9, pp. 1–14, 2017.
- [17] Y. Cheng and J. H. Hu, "Random error analysis and reduction for stochastic computation based on autocorrelation sequence," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 357–360, Melbourne, Australia, June 2014.



# Hindawi

Submit your manuscripts at  
<https://www.hindawi.com>

