Proceedings of the International Conference on Communication and Electronics Systems (ICCES 2018)
IEEE Xplore Part Number:CFP18AWO-ART; ISBN:978-1-5386-4765-3

Stochastic Computing: Systems, Applications, Challenges and Solutions

Romisa Akhtar
Department of Electronics and Instrumentation
University of Kashmir
Srinagar, India
romisaakhtar@gmail.com

Farooq A. Khanday
Department of Electronics and Instrumentation
University of Kashmir
Srinagar, India
farooqkhanday@kashmiruniversity.ac.in

***Abstract*— Modern day applications demands high speed and robustness with small integration area and low power consumption. Due to the advancement in technology, the dimensions of transistors goes on reducing thereby makes it possible to incorporate more and more number of transistors on a chip. However, reduction of the size of transistor increases leakage currents hence power consumption and unreliability. To address these problems either transistor has to be replaced by other device or classical way of computing has to be replaced by an alternative computing technique. One such unconventional form of computing is Stochastic Computing. Stochastic computing is probabilistic in nature. While weighted binary computing provides high speed and accuracy at the cost of large integration area and huge hardware cost. Also, these computing techniques have small noise robustness. Stochastic computing is efficient in terms of integration area, hardware cost and has high noise robustness. While the speed of stochastic computing circuits is comparable to weighted binary computing systems for small applications, their accuracy is far less then weighted binary computing circuits. This paper describes the key concepts of stochastic computing, its variants, conversion process and stochastic circuits. Main challenges to stochastic computing and their possible solutions are also highlighted in this paper.**

***Index Terms*—Stochastic Computing, Binary Computing, Stochastic numbers, Fault Tolerance, Stochastic circuits, Robustness.**

I. INTRODUCTION

CONTINUOUS scaling of MOS transistors led to the introduction of novel nano-scale devices. However, due to the reliability problems (uncertainty in circuit behavior) with these devices, scientists are moving from conventional deterministic computing towards a different technique called stochastic computing which is probabilistic in nature. The idea of stochastic computing was originally given by von Neumann [1] in 1950's, but it neither developed theoretically nor was implemented practically. The technological advances in the field of electronics and computing led to two groups of researchers independently develop the concept of stochastic computing. One group from "Standard Telecommunications Laboratories" in England composed of B. R. Gaines and J. H. Andrea were working in the field of machine learning [2, 3]. Other group from university of Illinois in U.S.A was working on image processing systems [4, 5]. Both groups developed the basic concepts of stochastic computing and implemented a number of functions in stochastic manner. Hence, are credited with being the parents of stochastic computing.

When probabilistic laws are applied to digital logic blocks, it results into stochastic computing. Data in stochastic computing is represented by the sequences of 0's and 1's, which are random in nature. These sequences are referred to as stochastic numbers, where probability of seeing a 1 denotes the value of the number. For example, stochastic number 0.25 can be represented by following sequences: 0010, 1000, 00010001, 001001001000 etc. Therefore, value of stochastic number depends on the ratio of number of 1's to the total length of a sequence and not on the position of 1's. In addition to this, the length of the sequence need not to be fixed.

The main advantage of stochastic computing is that it enables the implementation of complex arithmetic functions by means of standard logic gates. For example, multiplication of two numbers can be performed by simple AND gate. In addition to this, stochastic computing circuits are error tolerant. However, low precision, low accuracy and low speed are the drawbacks of stochastic computing [6].

As of now, the application domain of stochastic computing is far less than the domain of binary weighted computing. It is limited to few specific applications like neural networks [7, 8, 9, 10], image processing [11], LDPC decoding [12], pattern recognition [13, 14]. Real time image processing, medical implants, data mining are some of the new applications where stochastic computing is being used.

This paper describes the key concepts of stochastic computing, its variants, conversion process and stochastic circuits. It also includes the main challenges to stochastic computing and their possible solutions. The paper is organized as follows: different variants of stochastic logic is discussed in Section II. Section III describes the conversion process. The brief idea of arithmetic operations is given Section IV. Challenges and their possible solutions are discussed in Section V. This paper is concluded in Section VI.

II. STOCHASTIC NUMBER FORMAT

Initially there were two formats of stochastic numbers, namely unipolar and bipolar formats defined by B. R. Gaines [2, 3]. In these formats the numerical representation was limited to [0 to 1] and [-1 to +1] respectively. Bipolar format of a stochastic number is deduced from its unipolar format by making a change of variable. The relationship between bipolar and unipolar format of a stochastic number is given as:

$$p* = 2p - 1 \quad (1)$$

where p^* is the value of the stochastic number in bipolar format and p is the value of number in unipolar format.

Later on two new formats of stochastic numbers were proposed called unsigned extended stochastic logic (UESL) and signed extended stochastic logic (SESL) [10], where the numerical representation was extended to [0 to +∞] and [-∞ to +∞] respectively. In UESL, the stochastic numbers are represented by the ratio of two numbers in unipolar format while as in SESL these are represented by ratio of two numbers in bipolar format. Stochastic numbers in UESL are given as:

$$x = p/q \quad (2)$$

where p and q are the stochastic numbers in unipolar format.

While the stochastic numbers in SESL are given as:

$$x* = p*/q* \quad (3)$$

where p^* and q^* are the stochastic numbers in bipolar format.

Advantage of UESL and SESL over unipolar and bipolar formats is that they are more robust to noise and hence are more error tolerant [15]. However, their drawback is the requirement of more hardware for conversion. Other formats of stochastic numbers are inverted bipolar (IBP) format and ratioed format. IBP is the inverse of bipolar format in which the value of a stochastic number is given as:

$$p^i = 1 - 2p \quad (4)$$

where p is the value of stochastic number in unipolar format. IBP is more convenient to use with spectral transforms. This format greatly simplifies the analysis and synthesis of circuits in spectral domain.

In ratioed format, the value of a stochastic number is represented by the ratio of 1's to 0's of the bit stream. The representation space of this format is [0 to +∞].

Let us consider an example in which binary number 00000101 (5) will be converted into above discussed formats of stochastic number:

For an eight bit binary number, the length of stochastic number (stochastic bit stream) for each format will be 2^8 = 256 bits.

In Unipolar format, the value of binary number 00000101 as a stochastic number will be p = 5/256 = 0.0195. The number of 1's in the bit stream will be equal to 5 while as the number of 0's will be 251.

In Bipolar, inverse bipolar and ratioed format, the same number will have the value equal to (5-251)/256 = -0.961, (251-5)/256 = 0.961, 5/251 = 0.019 respectively.

In UESL format, the binary number 00000101 is encoded using two stochastic numbers "p" and "q" in unipolar format where p = 0.195 and q = 0.039.

Similarly in SESL format, the same number can be represented by two stochastic numbers "p*" and "q*" in bipolar format where p* = -0.61 and q* = -0.922.

III. CONVERSION BLOCKS

In order to use any variant of stochastic logic, first it is necessary to convert a binary number into its corresponding stochastic number and vice versa. This conversion is achieved by two blocks one is called Binary to Pulse converter and other is called pulse to binary converter. Binary to pulse converter shown in Fig. 1, as the name suggests converts a binary number into its corresponding stochastic number. It consists of a random number generator and a comparator, which produces an output equal to 1 at each clock cycle when the binary number is greater than the generated random number. This block produces a sequence of random 0's and 1's over a finite period of time called evaluation period. Hence, it resembles an ideal Bernoulli's process where probability of generating a 1 is p_x and for 0, it is 1-p_x. The probability density function and probability distribution function for this block is given as:

$$f_X(x) = \begin{cases} \frac{1}{2^n} & for\ 0 \le X < 2^n \\ 0 & for\ X \ge 2^n \end{cases} \quad (5)$$

$$F_X(x) = P(x < b) = \begin{cases} \frac{b}{2^n} & for\ 0 \le b < 2^n \\ 1 & for\ b \le 2^n \end{cases} \quad (6)$$

Pulse to binary converter shown in Fig. 2 consists of a simple counter which counts the number of 1's in stochastic signal. The output of this block is governed by binomial distribution, which is given as:

$$P_N(X) = \binom{N}{X}. p^X.(1-p)^{N-X} \quad (7)$$

where p represents the probability associated with stochastic signal.

In stochastic computing, these blocks costs more then the actual processing circuits. For example, the multiplication of two numbers requires a single AND gate while as conversion blocks for this circuit requires multiple number of gates. It is reported that conversion circuits consume up to 80% of total area [16]. Hence, there is a need for optimization.

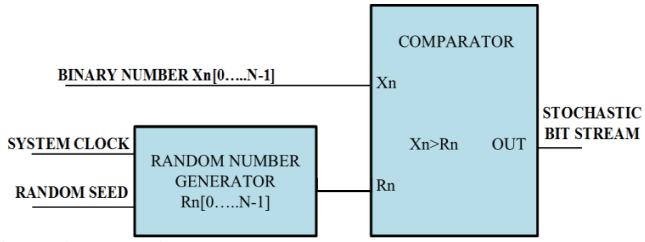

Fig. 1. Binary to Pulse converter.

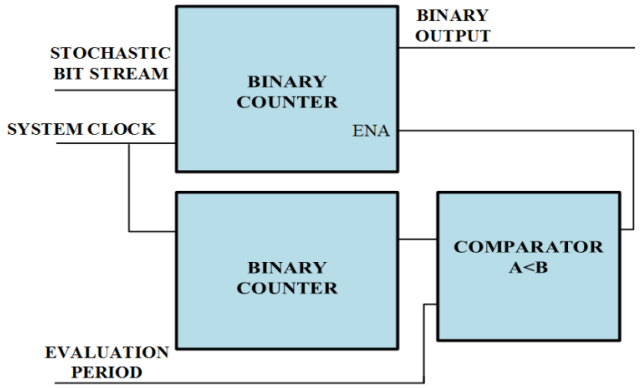

Fig. 2. Pulse to Binary converter.

IV. STOCHASTIC IMPLEMENTATION OF BASIC ARITHMETIC OPERATIONS

A. Complementary operation:

It is implemented in unipolar and bipolar format using a simple NOT gate shown in Fig. 3.

B. Multiplication Operation:

It is achieved by using a single AND gate in unipolar format, a single XNOR gate in bipolar format, two AND gates in UESL, two XNOR gates in SESL as shown in Fig. 4 [10]. 4–bit stochastic multiplier was proposed by Gupta and Kumaresan [17] in which linear feedback shift register was used as a random number generator and its results are highly accurate.

C. Addition Operation:

In unipolar as well as in bipolar format the addition operation is performed by a multiplexer as shown in Fig. 5. However, the output is the scaled sum of two inputs not the absolute sum. The absolute sum of two inputs is obtained in UESL and SESL format as shown in Fig. 6 and 7 respectively [10]. A number of adder designs are proposed in the literature like scaling free stochastic adder [18]. Another design of stochastic adder that eliminates the requirement of using separate random source is proposed by Lee et al. [19].

D. Subtraction Operation:

In unipolar format, subtraction operation is achieved by single XOR gate where the inputs to this gate must be correlated. However, the output is absolute difference and not the actual difference [20]. Figs. 8, 9, 10 and 11 shows the implementation of subtraction operation in unipolar, bipolar, UESL and SESL formats respectively [10, 20].

E. Division Operation

Fig. 12, 13 and 14 shows the division operation in unipolar, UESL and SESL formats respectively [2, 3, 10]. In unipolar and bipolar format, stochastic divider consists of Up/Down counter, binary to pulse converter, comparator, AND gate and XOR gate. Therefore, the hardware required for division operation in these formats is much more than in UESL and SESL, where it is achieved by only two AND gates and two XNOR gates respectively.

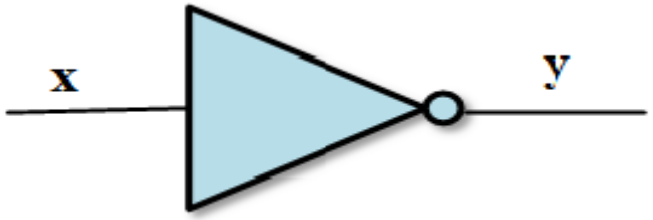

Fig. 3. Implementation of complementary function in unipolar and bipolar format.

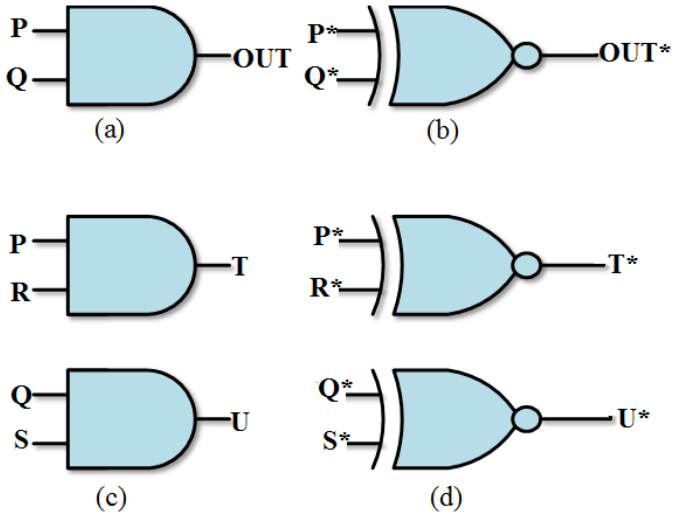

Fig. 4. Implementation of Multiplication operation in (a) unipolar format (b) Bipolar format (c) UESL format (d) SESL format.

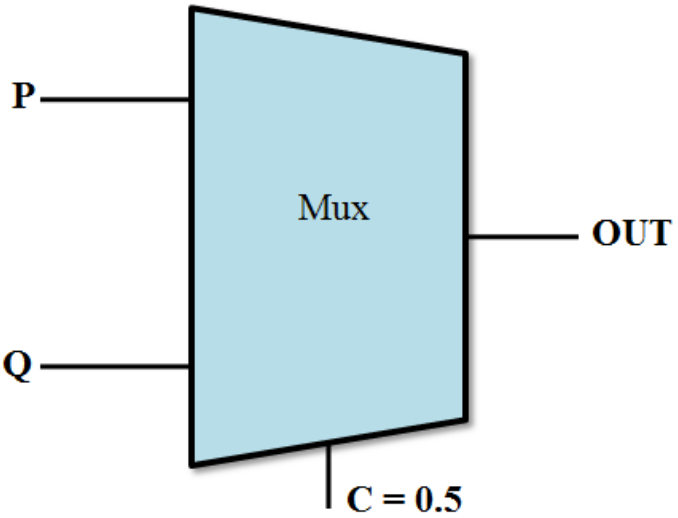

Fig. 5. Implementation of Addition operation in Unipolar and Bipolar format.

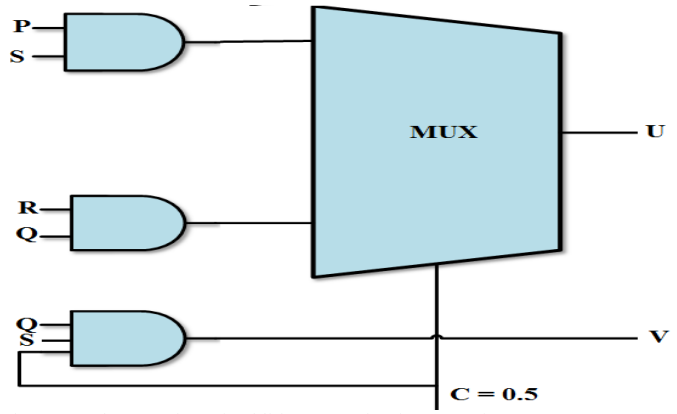

Fig. 6. Implementation of Addition operation in UESL format.

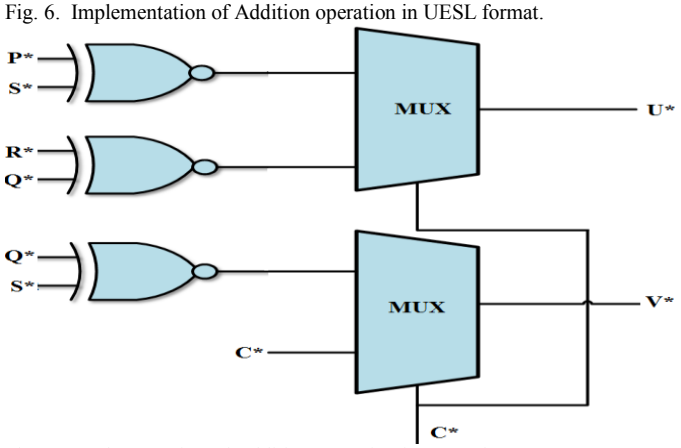

Fig. 7. Implementation of Addition operation in SESL format.

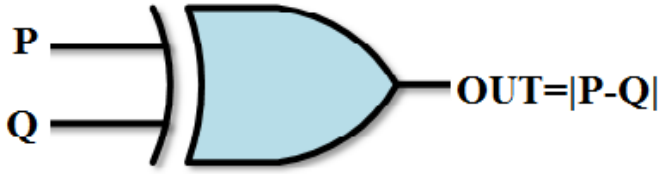

Fig. 8. Implementation of Subtraction operation in unipolar format.

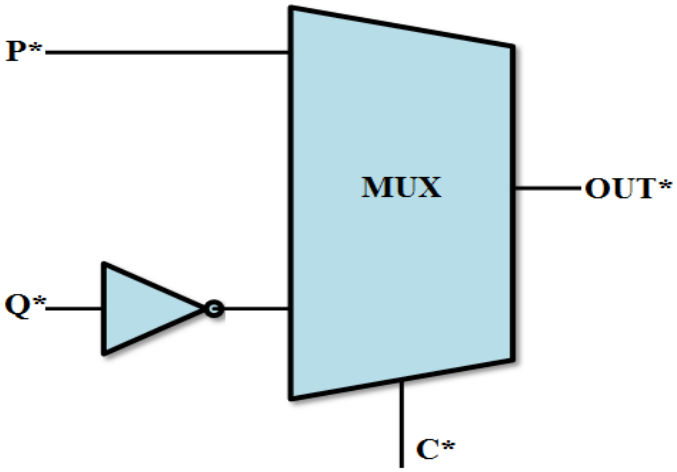

Fig. 9. Implementation of Subtraction operation in bipolar format.

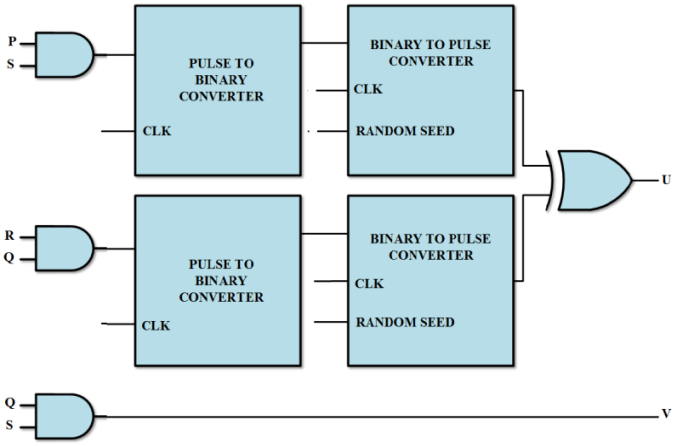

Fig. 10. Implementation of Subtraction operation in UESL format.

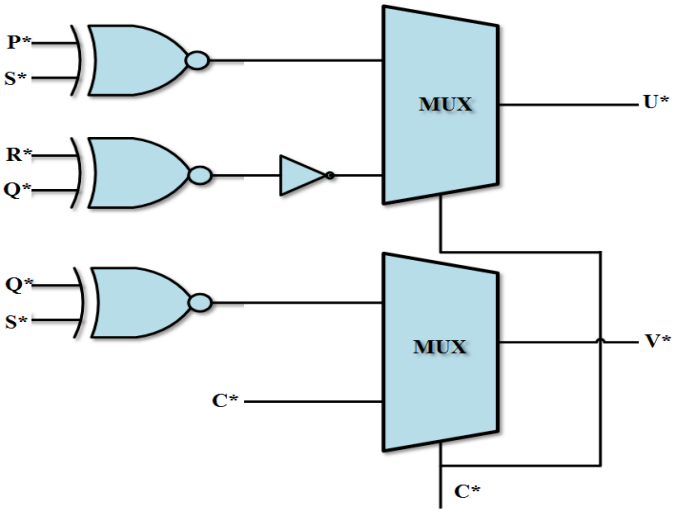

Fig. 11. Implementation of Subtraction operation in SESL format.

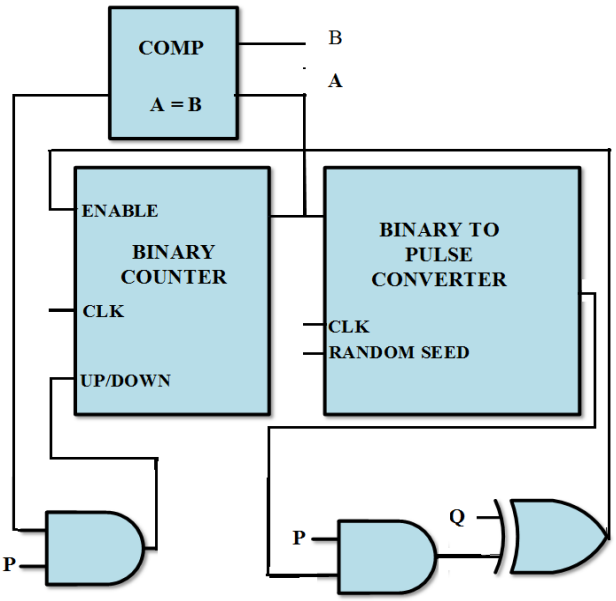

Fig. 12. Implementation of division in unipolar format.

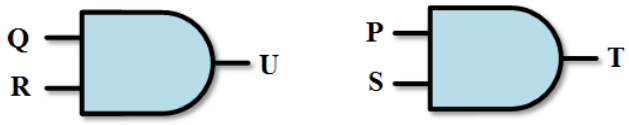

Fig. 13. Implementation of division in UESL format

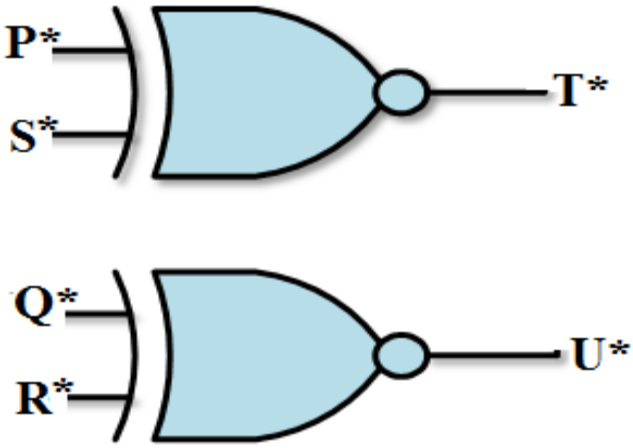

Fig. 14. Implementation of division in SESL format.

V. CHALLENGES

Conversion blocks of any stochastic circuit occupy 80% of the total area. In addition, the applications in which stochastic computing is used consists of a large number of inputs and outputs. Hence, there is a need for optimization of these blocks. The proposed optimized blocks are weighted binary generator [18], stochastic number generator by van Daalen et al [21] etc. For further optimization of these blocks chaotic circuits can be used as random number generator instead of linear feedback shift register [22].

As real world is analog in nature. In order to process real world signals with stochastic computing circuits, these signals need to be converted into binary format with the help of ADC's before being converted to stochastic format. This increases the overall cost of stochastic circuits. In order to reduce this cost, analog to stochastic converters were proposed which directly converts the analog signals into stochastic ones [23]. It also solves the problem of storing stochastic numbers in bigger systems such as instruction-set processors.

Main source of inaccuracy in stochastic computing is correlation among the signals. In order to get the accurate results, the inputs to the stochastic circuits must be uncorrelated, which requires extra hardware like different random number generators for different inputs. The sources of correlation in stochastic circuits is process of feedback where signal is fed back to the input resulting in autocorrelation and use of same random number generator for different inputs. As of now, a class of Boolean functions have been identified and analyzed called correlation insensitive functions [20]. An example of such function is the Boolean function of multiplexer. These functions are not influenced by the correlation among their inputs. Hence, same random number generator can be shared between different inputs.

Stochastic computing in its initial days lacked a comprehensive design method. This problem was solved by Armin et al. by proposing a synthesis algorithm called STRAUSS (Spectral transform use in stochastic circuit synthesis) [24, 25]. STRAUSS is based on spectral transform. According to this algorithm, a key relationship exists between Fourier transform of a Boolean function and the stochastic function implemented by it. STRAUSS designs combinational stochastic circuits, which are relatively easy to design, but suffers from a drawback of implementing a limited number of functions. In order to overcome this drawback sequential components are used in stochastic circuits. There are two classes of sequential stochastic circuits: UCB (Up/Down counter based) [2, 3, 8] and SRB (shift register based) [26, 27]. An algorithm called MOUSE (Monte-Carlo Optimization Using stochastic Equivalence) [28] is used for the optimization of SRB designs.

Stochastic computing is not an accurate computing technique. Inaccuracy in stochastic circuits can be due to correlation, dispersion associated with conversion process, random number source placement, lack of resolution in process of conversion and bit stream length. For example, in 2016, P.S.Ting et al cascaded two well-designed squarer circuits with input X, the implemented function turns out to be X^3 instead of X^4. In addition, there is a trade-off between accuracy and energy consumption. With increase in the bit stream length both energy and accuracy of the stochastic circuit increases. Therefore, if the application in which stochastic computation is used can tolerate small errors, bit stream length can be reduced in order to decrease energy consumption at the cost of less accuracy.

Length of bit streams increase exponentially with precision. For 8-bit precision, bit stream of length 256 is required. In order to increase the precision from 8 to 9 bits, the bit stream length has to be doubled. This increase in bit stream length results in long computation time. Hence, affecting the speed of the system.

V. CONCLUSION

Stochastic computing provides a low cost implementation of complex functions at the cost of accuracy. It is important to incorporate an "accuracy manager" in stochastic circuits, which will automatically adjust the design parameters affecting accuracy. Stochastic computing can work in different number formats. These number formats can be explored for stochastic designs with high accuracy, less complexity and less power consumption. Besides, a continuous research is being carried out to enhance the efficiency of stochastic circuits and exploring the new application areas of stochastic computing.

REFERENCES

[1]. J. von Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," Automata Studies, Princeton Univ. Press, pp. 43–98, 1956.

[2]. B. R. Gaines, "Stochastic computing," Proc. AFIPS Spring Joint Computer Conf., pp. 149–156, 1967. doi:10.1145/1465482.1465505.

[3]. B. R. Gaines, "Stochastic computing systems," Advances in Information Systems Science, pp. 37–172, 1969. doi:10.1007/978-1-4899-5841-9 2.

[4]. W. J. Poppelbaum, C. Afuso and J. W. Esch, "Stochastic computing elements and systems," Proc. AFIPS Fall Joint Computer Conf., pp. 635–644,1967. doi:10.1145/1465611.1465696.

[5]. W. J. Poppelbaum, "Statistical processors," Advances in Computers, pp. 187–230,1976. doi:10.1016/S0065-2458(08)60452-0.

[6]. The Promise and Challenge of Stochastic Computing Armin Alaghi, *Member, IEEE*, Weikang Qian, *Member, IEEE*, and John P. Hayes, *Life Fellow, IEEE.*

[7]. S.L. Bade and B.L. Hutchings, "FPGA-Based Stochastic Neural Networks-Implementation," *Proc. IEEE workshop on FPGAs for Custom Computing Machines*, pp. 189-198, 1994.

[8]. B.D. Brown and H.C. Card, "Stochastic Neural Computation I: Computational Elements," *IEEE Trans. Comp.*, vol. 50, pp. 891-905, 2001.

[9]. B.D. Brown and H.C. Card, "Stochastic Neural Computation II: Soft Competitive Learning," *IEEE Trans. Comp.*, vol. 50, pp. 906-920, 2001.

[10]. V. Canals, A. Morro, A. Oliver, M.L. Alomar and J.L. Rossello, "A New Stochastic Computing Methodology for Efficient Neural Network Implementation," *IEEE Trans. Neural Networks and Learning Systems*, vol. 27, pp. 551-564, 2016.

[11]. A. Alaghi, C. Li and J. P. Hayes, "Stochastic circuits for real-time image-processing applications," Proc. DAC, pp. 136:1–136:6, 2013. doi:10.1145/2463209.2488901.

[12]. W. J. Gross, V. C. Gaudet and A. Milner, "Stochastic implementation of LDPC decoders," Proc. Asilomar Conf. Signals, Systems and Computers, pp. 713–717, 2005. doi:10.1109/ACSSC.2005.1599845.

[13]. J.L. Rossello, V. Canals, I. de Paul, J. Segura, "Using Stochastic Logic for Efficient Pattern Recognition Analysis", IEEE International Joint Conference on Neural Networks(IJCNN 2008) (IEEE World Congress on Computational Intelligence), pp. 1057-1061, Hong Kong (China), June 2008.

[14]. V. Canals, A. Morro, J.L. Rosselló, "Stochastic-based pattern-recognition analysis", Pattern Recognition Letters, Volume 31, Issue 15, pp. 2353-2356, 2010.

[15]. V. Canals, A. Oliver, M.L. Alomar, M. Roca, E. Isern, E. Garcia-Moreno, A. Morro, F. Galán, J. Font-Rosselló and J.L. Rosselló,"Robust Stochastic Logic for Pattern Recognition",2016.

[16]. W. Qian, X. Li, M. Riedel, K. Bazargan and D.J. Lilja, "An Architecture for Fault-Tolerant Computation with Stochastic Logic," *IEEE Trans. Comp.*, vol. 60, pp. 93-105, 2011.

[17]. P. K. Gupta and R. Kumaresan, "Binary multiplication with PN sequences," IEEE Trans. Acoustics, Speech and Signal Processing, 36, 4, pp. 603–606, 1988. doi:10.1109/29.1564.

[18]. B. Yuan, Y. Wang and Z. Wang, "Area-Efficient Scaling-Free DFT/FFT Design Using Stochastic Computing," *IEEE Trans. Circuits and Systems II: Express Briefs*, vol. 63, pp. 1131-1135, 2016.

[19]. V.T. Lee, A. Alaghi, J.P. Hayes, V. Sathe and L. Ceze, "Energy-efficient hybrid stochastic-binary neural networks for near-sensor computing," *Proc. Design, Automation and Test in Europe Conf.* (DATE), pp. 13-18, 2017.

[20]. A. Alaghi and J.P. Hayes, "Exploiting Correlation in Stochastic Circuit Design," *Proc. Intl Conf. on Computer Design* (ICCD), pp. 39–46, Oct. 2013.

[21]. M. van Daalen et al., "Device for generating binary sequences for stochastic computing," Electronic Letters 29, 1, pp. 80-81, 1993. doi:10.1049/el:19930052.

[22]. Rosselló, J.L., Canals, V., de Paúl, I., Bota, S., Morro, A. "A Simple CMOS Chaotic Integrated Circuit", IEICE Electronics Express, Vol. 5, 1042–1048, 2008.

[23]. N. Onizawa et al., "Analog-to-stochastic converter using magnetic-tunnel junction devices," Proc. NANOARCH, pp. 59–64, 2014. doi:10.1109/NANOARCH.2014.6880490.

[24]. A. Alaghi and J. P. Hayes, "A spectral transform approach to stochastic circuits," Proc. ICCD, pp. 315–312, 2012. doi:10.1109/ICCD.2012.6378658.

[25]. A. Alaghi and J. P. Hayes, "STRAUSS: spectral transform use in stochastic circuit synthesis," IEEE Trans. on CAD, 2015. doi:10.1109/TCAD.2015.2432138.

[26]. Chen, T.-H. and Hayes, J.P. "Analyzing and controlling accuracy in stochastic circuits." *Proc. ICCD*, pp.367-373, 2014.

[27]. Ting, P.-S. and Hayes, J.P. "Isolation-based decorrelation of stochastic circuits." *Proc. ICCD*, pp.88-95, 2016.

[28]. Paishun Ting and John P. Hayes, " On the Role of Sequential Circuits in Stochastic Computing**",** 2018.

[29]. P. S. Ting and J. P. Hayes, "Isolation-Based Decorrelation of Stochastic Circuits," *Proc. Intl. Conf. Computer Design (ICCD)*, pp. 88-95, 2016.

[30]. F. Neugebauer I. Polian and J. P. Hayes, "Framework for Quantifying and Managing Accuracy in Stochastic Circuit Design," *Proc. Design, Automation and Test in Europe Conf.* (DATE), pp. 1-6, 2017.