Taking Advantage of Correlation in Stochastic Computing

Rahul Kumar Budhwani, Rengarajan Ragavan, Olivier Sentieys

Taking Advantage of Correlation in Stochastic Computing

Rahul Kumar Budhwani, Rengarajan Ragavan, Olivier Sentieys
IRISA/INRIA, University of Rennes, France
{rahul.budhwani,rengarajan.ragavan,olivier.sentieys}@irisa.fr

***Abstract*—In recent years, shrinking size in integrated circuits has imposed a big challenge in maintaining the reliability in conventional computing. Stochastic computing has been seen as a reliable, low-cost, and low-power alternative to overcome such issues. Stochastic Computing (SC) computes data in the form of bit streams of 1s and 0s. Therefore, SC outperforms conventional computing in terms of tolerance to soft error and uncertainty at the cost of increased computational time. Stochastic Computing with uncorrelated input streams requires streams to be highly independent for better accuracy. This results in more hardware consumption for conversion of binary numbers to stochastic streams. Correlation can be used to design Stochastic Computation Elements (SCE) with correlated input streams. These designs have higher accuracy and less hardware consumption. In this paper, we propose new SC designs to implement image processing algorithms with correlated input streams. Experimental results of proposed SC with correlated input streams show on average 37% improvement in accuracy with reduction of 50-90% in area and 20-85% in delay over existing stochastic designs.**

I. Introduction

As the device size is shrinking continuously, circuits are getting more vulnerable to soft errors generated in the circuit. To make circuit error-free, additional components are added in the circuit leading to increased hardware complexity, power consumption, and energy. This is expected to become more problematic with the increasing trend of technology scaling. Therefore, stochastic computing is seen as an alternative to conventional computing. SC designs require less hardware and are more tolerant to soft errors as compared to conventional computing at the expense of high latency. SC, initially introduced by Gaines [1], uses a probabilistic model of computation and requires less hardware to implement complex operations such as multiplications, etc. Basic stochastic elements and architectures are needed to implement stochastic computing. Stochastic numbers are represented by bit streams of 1s and 0s where the number of 1s in the stream determine the probability function depending upon representation. Following Equations 1 and 2 represent uni-polar and bipolar coded format of stochastic streams, respectively. Uni-polar and bipolar formats are used for positive and positive-or-negative fractional numbers, respectively.

$$P(1)_{stream} = p, \quad \text{if } 0 \leq p \leq 1 \tag{1}$$

<01001010> = 3/8

$$P(1)_{stream} = (p+1)/2, \quad \text{if } -1 \leq p \leq 1 \tag{2}$$

<01001000> = -1/2

$$f(x) = x^8 + x^6 + x^5 + x^4 + 1 \tag{3}$$

Fig. 1 shows the basic Stochastic Computation Elements (SCE) used in stochastic circuits with uncorrelated input streams [2]. Stochastic streams are generated using Stochastic Number Generators (SNG) shown Fig. 1(d). The (pseudo) random number generator (RNG) in SNG is implemented using Linear Feedback Shift Register (LFSR) [1]. We use 8-bit LFSR in this study to generate 256 bits of stochastic bitstream. The feedback polynomial of the 8-bit LFSR is shown in Equation 3.

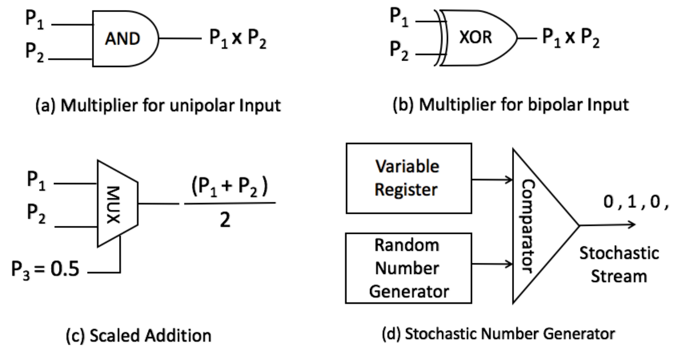

Fig. 1. Basic Stochastic Elements with uncorrelated input streams

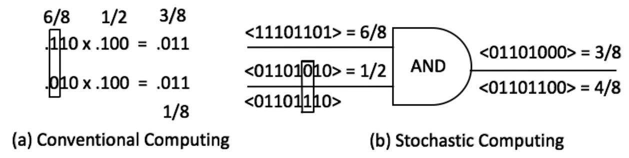

Fig. 2. Soft Error Effect on Conventional and Stochastic Computing

The main advantage of stochastic computing is its improved tolerance towards soft errors [3]. Fig. 2 shows the effect of bit flip due to soft error on the output of both conventional and stochastic computing. Because every bit flip in stochastic stream carries equal weight equivalent to the minimum change in the binary value, impact of soft error is less in SC compared to conventional computing. Secondly, requirement of less complex hardware in SC provides scope for massive parallelism. Finally, SC consumes less power due to reduced hardware complexity. These advantages come at the cost of high latency and reduced accuracy, making SC suitable for applications that can tolerate errors of acceptable margin. In SC, accuracy of the output depends on the correlation among the streams and the length of the stream. For higher accuracy, input streams are supposed to be highly independent and infinitely long as the accuracy increases as square root of n, where n is the stochastic stream length [2]. Due to the need of uncorrelated streams for better accuracy result, lot of hardware is used in the conversion of binary numbers to stochastic streams [2]. Therefore, there is a need to look for an alternate design to increase the accuracy without increasing the bit length. In our study, we have exploited correlation to increase the accuracy of stochastic computation. In this work we propose wise use of correlation to implement higher accuracy SCE designs with less hardware compared to existing SCE designs [4]. As a proof of concept, different image processing algorithms are implemented using the proposed SCE designs with correlated input streams.

The paper is organized as follows. Section 2 discusses how correlation can be used to implement high accuracy SCE. Section 3 discusses proposed stochastic designs for different image processing algorithms with correlated input streams. Section 4 compares the conventional computing with the proposed stochastic designs in terms of power, delay, area, accuracy, and fault tolerance. Finally, Section 5 provides conclusion and perspectives.

II. Correlation

In the existing stochastic methods, substantial amount of hardware is used to convert binary numbers to uncorrelated stochastic streams. In this paper, we show that the correlation can be exploited wisely to overcome such issues. [5], [6] and [7] provide an approach to generate correlated stochastic streams and its use in stochastic logic are discussed. This paper illustrates the use of stochastic logical blocks with correlated stochastic input streams to implement image processing algorithms. Some designs like Robert operator [8] have already been implemented using correlated stochastic streams. However, the Robert operator does not provide any information regarding the orientation of edge detection. Therefore, this paper further explores an alternative for an edge detection operator and other image processing algorithms.

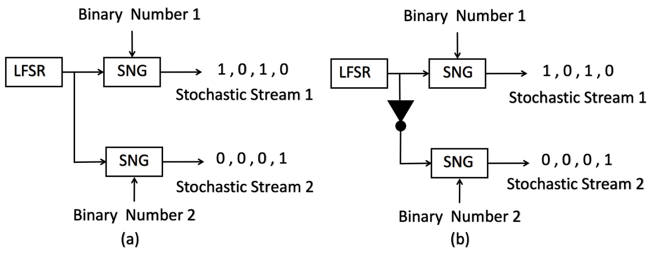

Fig. 3. (a) and (b) shows generation of positively and negatively correlated stochastic streams respectively

Fig. 3 shows the generation of positively and negatively correlated streams for stochastic computation. Fig. 4 shows the basic SCEs using correlated streams without any loss of accuracy [5]. We propose a modified scaled addition as shown in Fig. 4 (d), the LSB of LFSR is used as an input to select line of the MUX. This design avoids the need of an extra LFSR for selecting the mux line used in scaled adder [1]. We use these stochastic operators with correlated input streams to implement different image processing algorithms in the next section.

III. Stochastic Design of Image Processing Algorithms

In this section, we propose SC designs with correlated input streams for image processing algorithms. The algorithms used in this paper are median filter, contrast stretching, frame difference based image segmentation, and edge detection operator. We compare our proposed SC design of the algorithms with the existing SC and conventional computing designs.

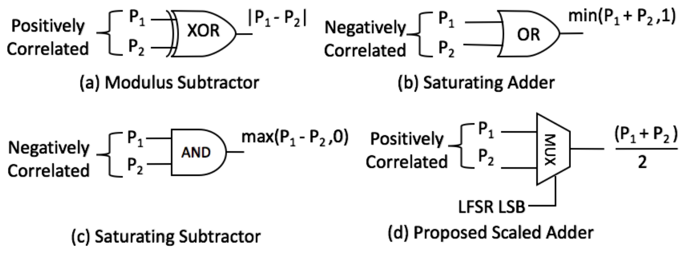

Fig. 4. (a), (b) and (c) shows existing stochastic operators with correlated inputs and (d) shows the proposed improvement in existing stochastic scaled adder

A. Median filter

Noise reduction based on the Median filter is a nonlinear digital filtering approach, mostly used to remove random noise like salt and pepper noise. This technique is widely used to reduce noise and preserve edges. In [4], an existing stochastic implementation of median filter is discussed and compared with conventional implementation. Fig. 5 shows the existing and proposed SC architecture for median filter, respectively. The proposed stochastic design is as accurate as the conventional design and eliminates the need for multiple LFSRs to generate independent streams prevalent in existing stochastic designs, which results in reduction of hardware cost.

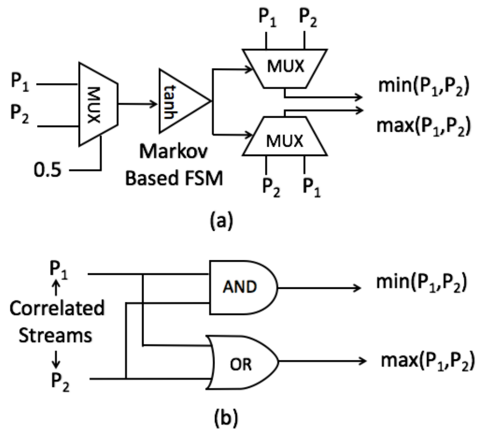

Fig. 5. (a) and (b) shows existing and proposed stochastic implementation of median filter, respectively

B. Contrast Stretching

Contrast stretching is an image enhancement technique used to improve the contrast of the image by stretching the intensity values in histogram to the full range of intensity values. The normalization of an intensity value is performed according to the piece-wise function defined in the stochastic domain as:

$$f(P_x) = \begin{cases} 0 & \text{if } P_x < \frac{a}{255} \\ \frac{x-a}{b-a} & \text{if } \frac{a}{255} \leq P_x \leq \frac{b}{255} \\ 1 & \text{if } P_x > \frac{a}{255} \end{cases}$$

Fig. 6 shows the proposed SC design of the above algorithm. In this design, we use CORDIV divider [9] for saturating division of two correlated stochastic streams. In Fig. 6, *a* and *b* represent lower and upper limits of pixel value for stretching respectively and *x* represents the pixel which is to be stretched.

The number of registers (K) in feedback path of the CORDIV divider decides the accuracy of the result. We have used 8 registers ($K = 8$) in our study for desirable performance and accuracy [9]. This design is also based on correlated input streams. Therefore, it consumes less hardware than the conventional stochastic design using independent stochastic streams and Markov-based FSM [4].

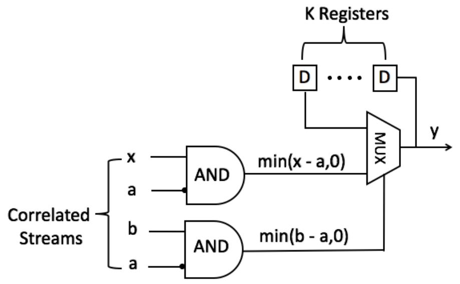

Fig. 6. Proposed stochastic implementation of contrast stretching algorithm

C. Frame Difference Based Image Segmentation

Frame segmentation is used to identify the differences between two frames/images. Frame segmentation highlights a particular pixel if the difference between the same pixel among two images is greater than a defined threshold value or else it masks that particular pixel. The piece-wise function for the frame difference based image segmentation in the probabilistic domain is:

$$f(X_t) = \begin{cases} 0 & \text{if } \frac{|X_{t+1} - X_t'|}{255} < \frac{X_{th}}{255} \\ 1 & \text{if } \frac{|X_{t+1} - X_t'|}{255} \geq \frac{X_{th}}{255} \end{cases}$$

where X$_t$ and X$_{t+1}$ is the value of the pixel at a particular location in current and next frame and X$_{th}$ is the pre-defined threshold pixel value. In [4], existing stochastic and conventional implementations are discussed for the above algorithm. The existing stochastic design uses the Markov-based FSM and multiple LFSRs for implementing the modulus function and generation of independent stochastic streams respectively. Fig. 7 shows the proposed SC design using correlated inputs consuming less hardware resources (logic and generation of stochastic streams) than existing stochastic designs.

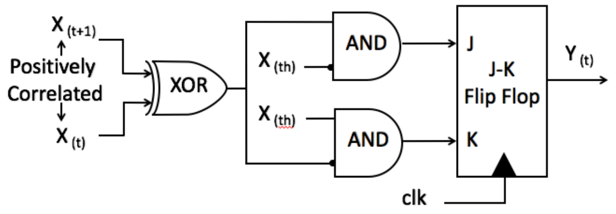

Fig. 7. Proposed stochastic implementation of frame based segmentation

D. Edge detection

The basic idea of edge detection is to detect changes in intensity for the purpose of finding an edge. The edge detection algorithm can be implemented by using first or second order derivatives. We propose a new first-order edge detection algorithm which is more accurate than the Robert algorithm proposed in [8]. The Robert algorithm does not provide any information regarding edge orientation. The proposed implementation of edge detection includes a 3×3 2D mask, with input pixels numbered from Z_1 to Z_9 which is used to calculate the intensity change for each pixel at center position, S_5. The intensity value at each pixel is calculated as

$$S_5 = \frac{1}{2}\left[\frac{1}{2}\left[\frac{1}{2}\Big[|Z_7 - Z_1| + |Z_8 - Z_2|\Big] + |Z_9 - Z_3|\right] + \frac{1}{2}\left[\frac{1}{2}\Big[|Z_3 - Z_1| + |Z_6 - Z_4|\Big] + |Z_9 - Z_7|\right]\right].$$

where Z_i represents value of the pixel at location i in a 3×3 window and S_5 is the pixel value at center location of the processed window. Fig. 8 shows the stochastic implementation of the proposed edge detection algorithm. All the stochastic streams are positively correlated and require only one LFSR for the generation of all the stochastic streams. For scaled addition, we have used different bits of the LFSR (Bit_1 to Bit_3 in Fig. 8) as select line for the multiplexer.

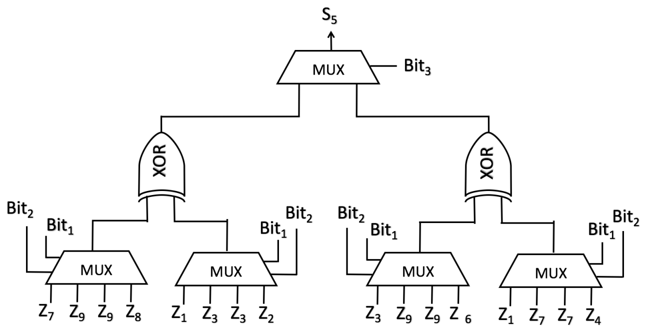

Fig. 8. Proposed stochastic implementation of edge detection algorithm

IV. Experiments and Results

In this section, we present the experimental results of conventional, existing, and proposed stochastic implementations of the image processing algorithms discussed in Section III. We have implemented these designs on an Xilinx ZYNQ 706 board and Xilinx Vivado tool flow is used to estimate the area and delay. Accuracy is obtained by simulation of the designs on different input data.

We used 256 bits of stochastic stream to represent a pixel intensity value. Table I shows the comparison of conventional, existing, and proposed stochastic implementations in terms of accuracy, area, and delay. For measurement of accuracy, we calculate mean output accuracy reduction per pixel as

$$E = \sum_{i=1}^{H} \sum_{j=1}^{W} \frac{|C_{i,j} - S_{i,j}|}{H \cdot W \cdot C_{i,j}}, \tag{4}$$

where H and W are height and width of the image respectively; C and S are the output pixel of the conventional and stochastic implementation of an algorithm respectively. Fig. 9 shows the simulation result of both conventional and proposed stochastic implementations of different image processing algorithms.

From Table I, it is evident that the accuracy of the proposed stochastic designs are better than the existing stochastic implementations. The proposed stochastic design for median filter provides maximum accuracy with lower hardware cost and lesser delay compared to existing SC design [4]. This reduction in hardware cost is achieved by replacing Markov-based FSM blocks for computation of uncorrelated bit streams

TABLE I. Comparison of conventional, existing, and proposed stochastic implementations in terms of accuracy, area, and delay

Benchmarks	Conventional Implementation			Existing Stochastic Implementation			Proposed Stochastic Implementation		
	Mean Accuracy reduction per pixel (%)	Area (LUTs)	Delay (ns)	Mean Accuracy reduction per pixel (%)	Area (LUTs)	Delay (ns)	Mean Accuracy reduction per pixel (%)	Area (LUTs)	Delay (ns)
Median Filter	0.00	234	15.98	1.82	478	5921.5	0.00	50	903.42
Contrast Stretching	0.00	291	24.04	4.96	42	921.08	3.11	22	573.44
Frame Segmentation	0.00	16	3.88	0.82	43	1062.91	0.52	21	860.16
Edge Detection	0.00	116	4.39	6.8	98	2361.6	4.25	45	767.23

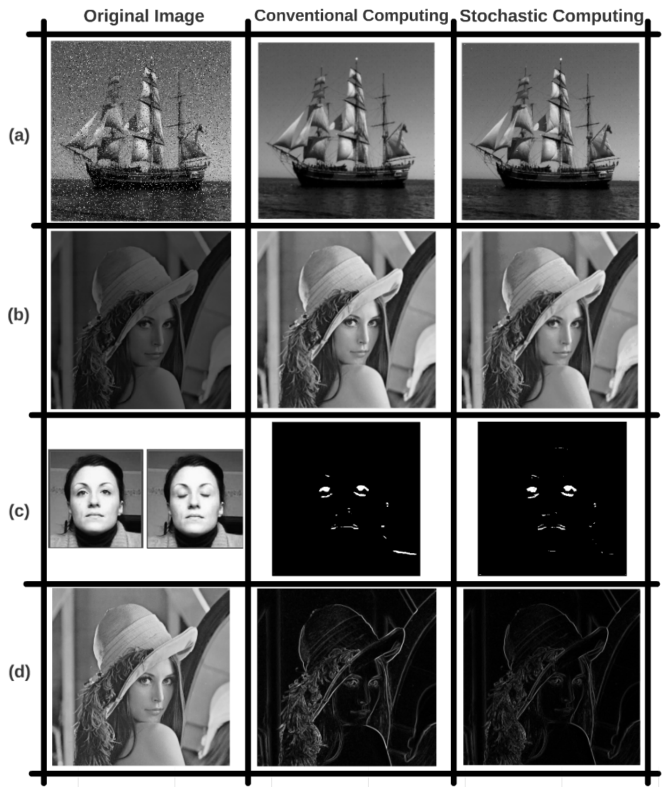

Fig. 9. (a), (b), (c) and (d) show simulation results of conventional and proposed stochastic implementation of median filter, contrast stretching, frame based image segmentation, and edge detection algorithm respectively

by simpler logic blocks like AND/OR/MUX gates with correlated bitstreams. In the proposed SC design of median filter, 38 Markov-based FSM blocks consuming higher area and latency are replaced by 38 AND/OR gate which results in reduction of 90% and 85% in area and delay respectively. Similarly, proposed SC design for other algorithms also shows reduction in area and delay compared to the existing SC designs. On average, the proposed stochastic designs for the algorithms show 37% improvement in accuracy, with significant reduction of 50-90% in area and 20-85% in delay as compared to existing stochastic designs in [4].

We have also studied the effect of soft error injection in both proposed stochastic and conventional implementations. We have used the methodology for soft error injection as stated in [4]. Table II shows the effect of 0%,10%, and 20% soft error injection in proposed stochastic and conventional implementations. Mean accuracy reduction per pixel due to effect of soft errors injection is 50-90% lesser in the proposed stochastic implementation as compared to conventional implementation.

TABLE II. Comparison of fault tolerance between conventional and proposed stochastic implementations

	Mean Accuracy reduction per pixel (%)					
	Conventional Implementation			Proposed Stochastic Implementation		
Soft Error	0%	10%	20%	0%	10%	20%
Median Filter	0.00	2.39	4.21	0.00	1.12	1.24
Contrast Stretching	0.00	10.42	18.69	3.11	6.81	9.69
Frame Segmentation	0.00	11.57	20.57	0.52	1.52	2.26
Edge Detection	0.00	8.76	18.48	4.25	5.12	7.26

V. Conclusion

Stochastic Computing is an intriguing alternative to conventional computing which requires less hardware resources and provides high tolerance towards soft errors. Existing stochastic designs are consuming more hardware resources for converting binary number to stochastic stream. In this paper, we have proposed stochastic designs with reduced hardware cost and delay by introducing correlation in the input streams. Correlation improves accuracy by 37% on average compared to the existing stochastic designs. Also significant reduction of 50-90% in area and 20-85% in delay are achieved in the proposed method. Correlation can therefore be used wisely to implement more efficient stochastic designs.

References

[1] B. R. Gaines, "Stochastic computing," in *Proc. of the spring joint computer conference*. ACM, 1967, pp. 149–156.

[2] A. Alaghi and J. P. Hayes, "Survey of stochastic computing," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 12, no. 2s, p. 92, 2013.

[3] 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.

[4] P. Li, D. J. Lilja *et al.*, "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.

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

[6] T.-H. Chen and J. P. Hayes, "Analyzing and controlling accuracy in stochastic circuits," in *Computer Design (ICCD), 2014 32nd IEEE International Conference on*. IEEE, 2014, pp. 367–373.

[7] M. Parhi, M. D. Riedel, and K. K. Parhi, "Effect of bit-level correlation in stochastic computing," in *Digital Signal Processing (DSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 463–467.

[8] A. Alaghi, C. Li, and J. P. Hayes, "Stochastic circuits for real-time image-processing applications," in *Proc. of the 50th Design Automation Conference*, 2013.

[9] T.-H. Chen and J. P. Hayes, "Design of division circuits for stochastic computing," in *VLSI (ISVLSI), 2016 IEEE Computer Society Annual Symposium on*. IEEE, 2016, pp. 116–121.