High-Speed Stochastic Circuits Using Synchronous Analog Pulses

M. Hassan Najafi and David J. Lilja
najaf011@umn.edu, lilja@umn.edu
Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis

Abstract— The primary advantages of stochastic computing are the very simple hardware required to implement complex operations, its ability to gracefully tolerate noise, and the skew tolerance. Its relatively long latency, however, is a potential barrier to widespread use of this paradigm, particularly when high accuracy is required. This work proposes a new, high-speed, yet accurate approach for implementing stochastic circuits that uses synchronized analog pulses as a new way of representing correlated stochastic numbers.

I. Introduction

Stochastic Computing (SC) [2] is a re-emerging computing paradigm first introduced by Gaines in 1969 [5]. In this paradigm, logical computation is performed on random bit streams called stochastic numbers (SNs). There are two well-known representations of SNs. In the unipolar representation, each real valued number x ($0 \leq x \leq 1$) is represented by a sequence of random bits, each of which has probability x of being one and probability $1 - x$ of being zero. In the bipolar representation (-1 $\leq x \leq$ 1), each bit in the stream has a probability $(x + 1)/2$ of being one and 1-$(x + 1)/2$ of being zero. For example, 10011, 10101, and 11100 are all SNs representing 0.60 in the unipolar and 0.2 in the bipolar representations.

The key advantages of stochastic computing are the very simple hardware required to implement complex operations [7], its ability to gracefully tolerate noise [11], and the skew tolerance in the arrival time of the computational units' inputs [10]. A significant challenge, however, to the widespread use of SC is the need for long stochastic bit streams and, therefore, long operation times, particularly if high accuracy is required [6]. While accuracy can always be sacrificed for the operation time by reducing the length of the bit streams, there is still a huge gap between the operation time of stochastic circuits and the conventional deterministic binary designs.

Independent (i.e., uncorrelated) bit streams are usually desired in stochastic circuits [2]. An AND gate multiplies two SNs only if its inputs are independent bit streams. However, correlated inputs can sometimes be advantageous. Correlation has been exploited in prior work in designing low-cost stochastic circuits. Alaghi *et al* [1] proposed low-cost stochastic circuits for the absolute-value subtraction, maximum, and minimum value functions that use correlated input streams. There is also a third class of operations that are insensitive to the correlation between inputs. A multiplexer (MUX) performs scaled addition/subtraction in the stochastic domain. Since at any time only one of the inputs is connected to the output, its performance is insensitive to the correlation between inputs.

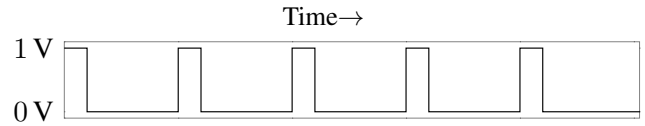


Fig. 1. A PWM signal with 20% duty cycle, representing 0.2 in unipolar and -0.6 in bipolar representations.

To move between the deterministic binary domain and the stochstic domain, a real-valued number must be converted into a corresponding SN. In prior work, SNs were often generated from pseudo-random constructs such as LFSRs. In each clock cycle, an unbiased random value r from the pseudo-random source was compared to a target value x. The output is a one if $r \leq x$ and a zero otherwise. Stochastic streams corresponding to different input values will be correlated if the same sequence of random values is used to convert the input values. The ones in the generated streams will be highly overlapped in such cases.

The representation of SNs is not limited to only digital bits. An analog interpretation of SNs has been previously proposed [10] by encoding the value as the fraction of time the signal is high. In this paper, we use a novel approach for generating correlated stochastic signals using analog periodic pulse signals. By exploiting pulse width modulation (PWM), stochastic signals with specific probabilities can be generated by adjusting the frequency and duty cycles of the PWM signals. These signals can be treated as inputs to the stochastic computation, with the value defined by the duty cycle. This observation is motivated by noting that the stochastic representation is a uniform, fractional representation. All that matters in terms of the value that is computed is the fraction of time that the signal is high [10]. For example, if a signal is high 20% of the time, it is evaluated as 0.20 in unipolar and -0.6 in bipolar representations, as shown in Fig. 1.

While there is still no general method for synthesizing stochastic operations to work on correlated inputs, we propose low-cost stochastic circuits that work on correlated inputs for two previously proposed expensive stochastic functions,

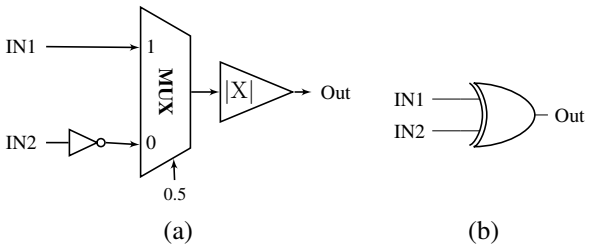


Fig. 2. Stochastic implementations of the absolute-valued subtraction function working on (a) both correlated and uncorrelated inputs [8], (b) only correlated inputs [1].

namely, the stochastic sort function and the stochastic comparator [9]. We replace the high-cost FSM-based circuits used in implementing the stochastic sort function with simple AND-OR gates and also propose a low-cost SN comparator using a single D-flip flop. Furthermore, as a solution to the long runtime problem of a specific class of stochastic circuits, namely circuits that work on correlated inputs, we propose to use synchronized analog PWM signals. By connecting synchronized, highly overlapped PWM signals to the inputs of these circuits, accurate outputs are ready right after performing the operations for only one period of the input signals. This is a time equal to one clock cycle in the conventional stochastic design.

II. Correlation in Stochastic Circuits

The inputs of a stochastic circuit must usually be independent or uncorrelated in order to achieve the desired functionality [2]. For most stochastic operations, correlation or similarity caused by insufficient randomness among input streams is often the main cause of inaccurate results. Alaghi and Hayes [1], however, suggested that correlation in SC is not always harmful. In fact, in some cases, correlated inputs can change the functionality of a circuit, which might result in a more desirable operation.

The XOR gate provides the most intuitive example of a case where correlation changes the functionality of the circuit. If an XOR gate is implemented under the usual assumption of independent inputs, it computes $x_1(1-x_2)+x_2(1-x_1)$. However, when it is connected with highly correlated inputs where the two input streams have maximum overlap in their 1s, it implements an entirely different function, namely, absolute-valued subtraction $|x_1 - x_2|$ [1]. This function was implemented conventionally using a combination of a NOT gate, a MUX, and a relatively high-cost FSM-based stochastic absolute value circuit (Fig. 2(a)) [8]. Evidently, when correlation is exploited, a much cheaper implementation of this function using just a single XOR gate is achievable, as seen in Fig. 2(b). As an example, connecting S1=11101 and S2=10001, two correlated stochastic streams representing 4/5 and 2/5, to the inputs of an XOR gate produces S3=01100, the expected value from performing absolute-valued subtraction.

The maximum and minimum value functions are two more examples of functions for which their stochastic implementations are significantly simpler when the input streams are correlated. If an AND gate is implemented under the usual assumption of independent inputs, it works as a multiplier. However, when it is connected with highly correlated inputs, it gives the minimum of the two stochastic streams. An OR gate, on the other hand, gives the maximum value of two SNs when its inputs are fed with highly correlated streams.

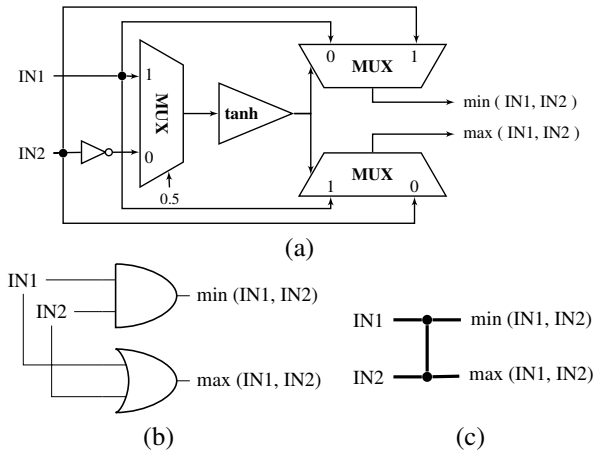


Fig. 3. Basic sorting unit: (a) conventional stochastic implementation working on both independent and correlated bipolar inputs [9]; (b) the proposed low-cost stochastic implementation working on correlated inputs (unipolar and bipolar); (c) the simplified model.

Fig. 3 shows two different stochastic implementations of a basic sorting unit to find the minimum and maximum values between two SNs. While the circuit presented in Fig. 3(b) is a low-cost circuit working only on highly correlated inputs, circuit (a) is the high-cost stochastic implementation previously proposed in [9] which works on both independent and correlated SNs. The FSM-based stochastic $tanh$ function along with the three MUXs makes circuit (a) much more expensive than circuit (b).

An ad-hoc method for synthesizing stochastic operations with correlated inputs is introduced in [1]. However, not all operations are synthesizable with current methods. Finding a comprehensive method for synthesizing any stochastic operations to work with correlated inputs is a work in progress. In this paper, we define correlation in SNs that are represented by analog signals. We show that, by using synchronous analog pulses as the inputs to stochastic circuits, not only do we still have the area saving benefits of correlated stochastic design, but we also can eliminate a major source of inaccuracy in SC, specifically, the random fluctuation inherent in generating SNs. Highly accurate results are ready after performing the operations on only one period of the analog periodic pulse signals.

III. Stochastic Circuits with Synchronized PWM Signals

When representing SNs with PWM signals, high correlation or maximum overlap can be provided by satisfying two requirements: 1) choosing the same frequency for the signals, and 2) having maximum overlap between the high parts of the signals. For example, two PWM signals that have the same period, and have the high part in each one located at the beginning

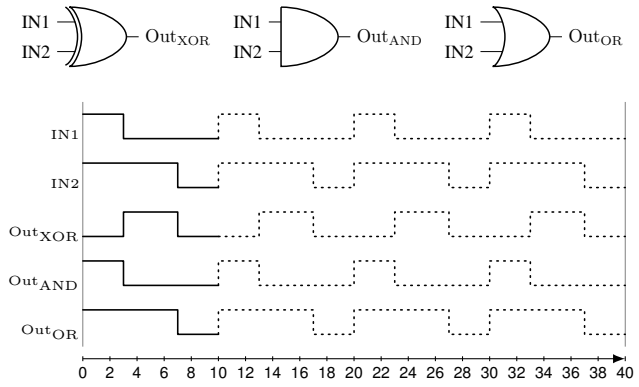


Fig. 4. Examples of performing stochastic absolute-valued subtraction, minimum, and maximum operations on two synchronized PWM signals: IN1 represents 0.3 and IN2 represents 0.7 . Both PWM signals have a period of 10ns. The output signals represent 0.4, 0.3, and 0.7, the expected values for the stochastic operations.

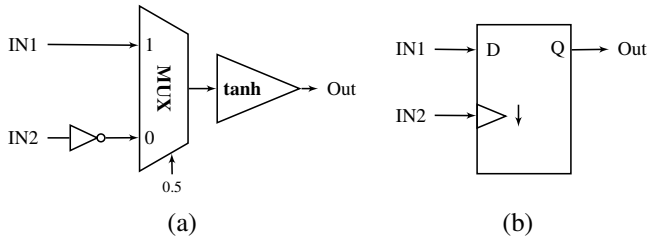


Fig. 5. (a) High-cost stochastic comparator proposed in [8], (b) new low-cost stochastic comparator.

or end of each period, are called correlated or synchronized signals. Fig. 4 shows two synchronized PWM signals and the outputs of performing the stochastic absolute-valued subtraction, minimum, and maximum operations on them. Obviously, the expected output is ready right after one cycle of the PWM input signals. Furthermore, continuing the operations for additional cycles does not improve the accuracy of the results.

Comparison of SNs is another common operation in stochastic circuits. Li and Lilja [8] proposed a stochastic comparator using the FSM-based stochastic $tanh$ circuit developed by Brown and Card [4], shown in Fig. 5(a). However, FSM-based circuits are often very expensive to implement. Fig. 5(b) shows how to use a simple D-type flip-flop as a stochastic comparator. For correct functionality, the inputs of the flip-flop must be correlated. For a digital representation, all 1s in each stream must be placed together at the beginning of the stream; for example, 11100 to represent 0.6. For the analog encoding, there must be only one high pulse in each period while the high part of the different signals must start at the same time to satisfy the correlation requirements. The first SN should be connected to the D input and the second one should be connected to the falling edge triggered clock input. The output of comparing two stochastic input numbers, N1 and N2, will be:

if (IN1 $<$ IN2) then Out=0; else Out=1

Fig. 6 shows two possible cases of comparing SNs represented by PWM signals using the proposed comparator. When IN1 is smaller than IN2, the falling edge of the PWM signal representing N2 causes the flip-flop to sample a low level signal, and so logical-0 is produced at the output. When N1 is greater than N2, the PWM signal representing N1 is still at a high level when the falling edge of IN2 occurs. So, logical-1 will be produced at the output of the flip-flop. In contrast to an approximate 0 and 1 produced at the output of the high-cost stochastic comparator in Fig. 5(a), the output of comparing the PWM stochastic signals using the flip-flop is a perfect representation of the 0 and 1 value.

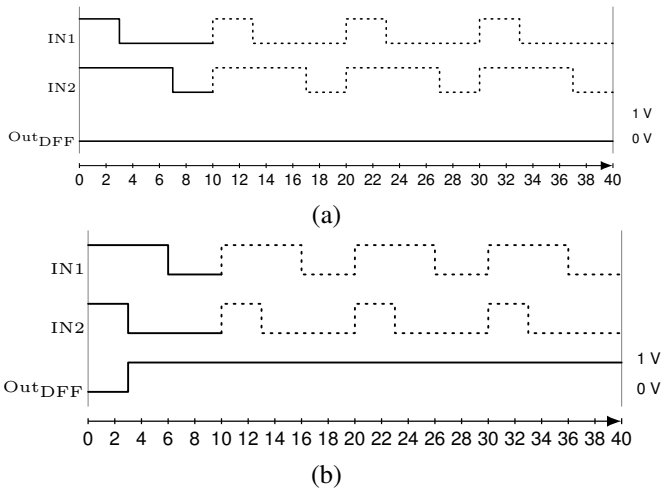


Fig. 6. Two examples of comparing SNs, represented by correlated PWM signals, using the proposed stochastic comparator: (a) IN1 $<$ IN2 and so Out=0, (a) IN1 $>$ IN2 and so Out=1.

As can be seen in Figures 4 and 6, the fraction of the time each one of the output signals is high is the same in all periods of each output signal. This observation concludes the main advantage of exploiting synchronized PWM signals in performing stochastic operations: the output of performing stochastic operations on synchronous analog pulses is ready after running the operation for only one period of the input signals. In such cases continuing the operation for additional periods does not change the value or, most importantly, the accuracy of the output. Operations performed on conventional digital bit-streams require running the operations for a large number of clock cycles to produce an accurate output.

IV. Case Studies

In this paper, we use the stochastic implementations of three digital image processing algorithms as case studies to evaluate the idea of using synchronous analog pulses as SNs and compare the area-latency benefits of exploiting correlation in stochastic circuit design.

A. Edge Detection

Fig. 7 shows two stochastic implementations of the Robert's cross edge detection algorithm, one working on both correlated and uncorrelated inputs (recall that the MUX is correlation insensitive), and the other working only on correlated inputs. While there is a slight difference between the functions implemented by these circuits, both functions can be decomposed

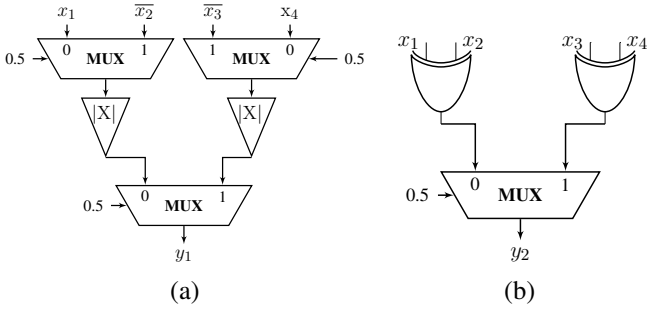


Fig. 7. Stochastic implementation of the Robert's cross edge detection algorithm working on (a) both correlated and uncorrelated stochastic inputs [9]; (b) only correlated stochastic inputs [3]. Note that the inputs and output of circuit (a) must be in bipolar format while circuit (b) supports both unipolar and bipolar representations.

into two absolute-valued subtraction and one scaled addition operations, as shown in these equations:

$$y_1 = 0.5 \times (0.5 \times |x_1 - x_2| + 0.5 \times |x_3 - x_4|)$$
$$y_2 = 0.5 \times (|x_1 - x_2| + |x_3 - x_4|)$$

Since the circuit presented in Fig. 7(a) uses two high-cost FSM-based stochastic absolute value circuits, its implementation cost is about 20 times more expensive than the circuit presented in Fig. 7(b) [3].

When PWM signals are going to be used as the inputs of circuit (b), the signals connected to each XOR gate should be synchronized. For the select input of the MUX we need a PWM signal with a duty cycle of 50% and a period smaller than the period of the input signals. By connecting a high frequency clock signal to the select input of the MUX, the output of the circuit is ready after processing the inputs for only one period of the PWM input signals.

B. Noise Reduction

The median filter replaces each pixel of an input image with the median of neighboring pixels. It is quite popular because, for certain types of random noise, it provides excellent noise-reduction capabilities. A hardware implementation of the 3x3 median filter based on a sorting network is shown in Fig. 8 [9]. A basic sorting unit is conventionally implemented with the stochastic circuit presented in Fig. 3(a). Peng *et al* [9] reported a hardware resource requirement of 125,000 NAND gates to implement the stochastic 3x3 median filter circuit. Here, we propose a much less expensive implementation by exploiting correlated inputs.

With all of the inputs stochastically correlated, we use the low-cost circuit presented in Fig. 3(b) for the sorting units. With this sorting circuit, the hardware cost is reduced to only 30 gates (15 AND + 15 OR). Similar to the edge detection circuit, highly overlapped PWM signals (or correlated bit-streams) must be used as the inputs of the proposed circuit. Measuring the fraction of the time the produced output signal is high for only one period of the input signals gives a highly accurate approximation of the median value. We will show that this processing time is much less than the latency when conventional stochastic bit-streams are used.

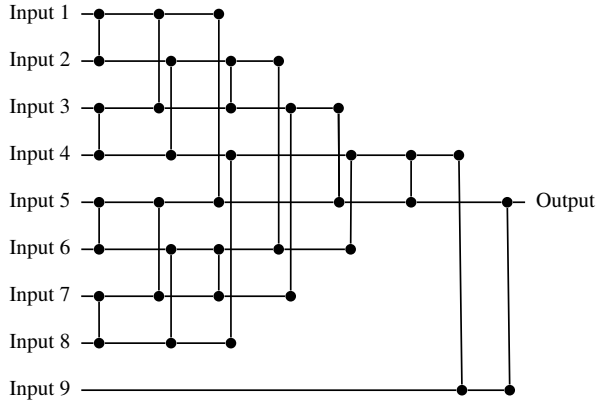


Fig. 8. Hardware implementation of the 3x3 median filter based on a sorting network [9].

C. Image Segmentation

The frame difference-based image segmentation algorithm is an image processing method which uses the difference between the value of a pixel at the current frame as X_t, and the value of the pixel at the same location of the previous frame as X_{t-1} to determine if it is greater than a predefined threshold, TH. If the difference is greater than TH, the pixel is set to the foreground value or 1; otherwise, it is set to the background value or 0. Li *et al* [9] proposed an expensive stochastic implementation for this algorithm using two NOT gates, two MUXs, and two high-cost FSM-based circuits for implementing the required absolute value and tanh functions, as shown in Fig. 9(a). The inputs to this circuit must be in the bipolar format while the output must be read as a unipolar number. Since the MUXs implement scaled subtraction, and since the stochastic tanh function leads to poor comparison results when its input is less than 0.2 [11], the circuit presented in Fig. 9(a) shows relatively poor performance for consecutive video frames and small TH values.

Exploiting the low-cost comparator proposed in this paper, we propose an accurate low-cost stochastic circuit for the frame difference-based image segmentation algorithm. The proposed circuit is shown in Fig. 9(b). The key to correct functionality of this circuit is to have highly correlated inputs. These can be correlated bit-streams or synchronized PWM signals.

Fig. 10 shows the performance of the proposed circuit using three sample inputs. When the signal produced at the output of the XOR gate, which computes |IN1-IN2|, goes high, it enables the SN generator responsible for converting the TH value. As a result, the XOR's output and the generated signal will be highly overlapped. Connecting two highly overlapped PWM signals to the inputs of a D-flip flop, the output is the result of comparing their corresponding values. The signal at the output of the D-flip flop is the expected output value from the frame difference-based image segmentation algorithm. Note that for this circuit the process of measuring the output must be delayed until the falling edge of TH signal. Then, instead of measuring the fraction of time the signal is high, the level of the output signal can be directly converted to a logical-0 or 1.

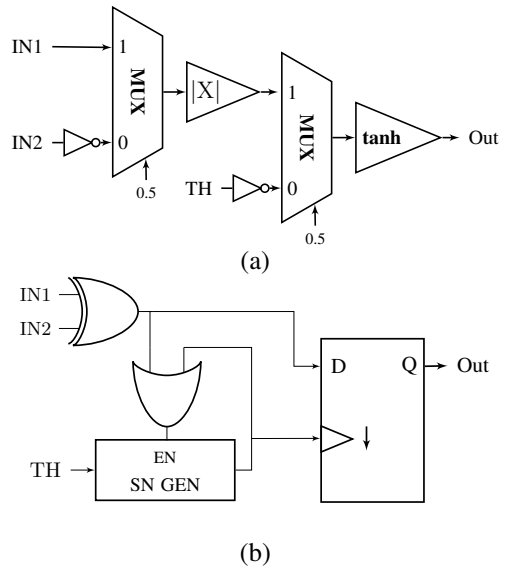


Fig. 9. (a) Conventional stochastic implementation [9] and (b) proposed low-cost stochastic implementation of the frame difference-based image segmentation algorithm.

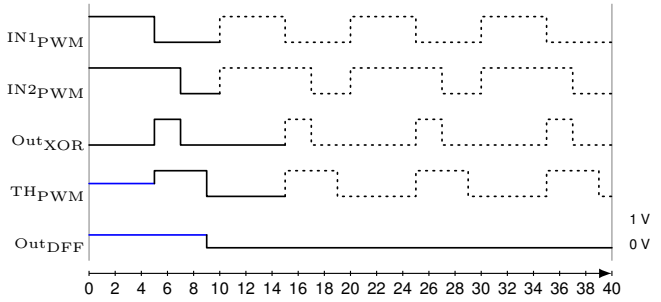


Fig. 10. Performance of the proposed frame difference stochastic circuit for three sample inputs: IN1, IN2, and TH are three PWM input signals with duty cycles 50%, 70%, 40%, and periods of 10ns.

V. Experimental Results

A hardware cost comparison of the stochastic circuits is shown in Table I. It can be seen that the circuits that are synthesized to work only with correlated inputs use substantially less hardware resources. To compare the operation time of the circuits, we synthesized them to determine the maximum feasible working frequency using the Synopsys Design Compiler vH2013.12 with a 45nm gate library. The critical path latency and the total latency of each circuit when operating for 256 clock cycles are presented in Table II. Clearly, the simple architecture of the circuits synthesized for the correlated inputs approach leads to a lower critical path and, consequently, a lower total latency required to obtain the final result.

To evaluate performance of the circuits, we simulated their operation with both independent and correlated input streams on the 128*128 Lena image for the edge detection circuit, on a 128*128 noisy soldier image for the noise reduction circuit, and on the two consecutive 144*144 frames of a walking man video for the frame difference circuit. The sample input images are shown in Fig. 11(a). The SN generator proposed in [12] was used for converting input pixel intensities into random bit-

TABLE I
Hardware resources required for the stochastic circuit design using independent input streams and the low-cost design using correlated inputs.

Case Study	Independent Stochastic [9]	Correlated Stochastic
Edge detection	110 NAND	2 NOT, 2 XOR, 1 MUX
Noise reduction	125k NAND	15 AND, 15 OR
Frame difference	107 NAND	1 XOR, 1 OR, 1 DFF

TABLE II
The critical path and total latency when processing one set of inputs with the conventional approach of generating stochastic streams.

Case Study	Independent Stochastic		Correlated Stochastic	
	CP	Latency	CP	Latency
Edge detection	0.39ns	99.8ns	0.30ns	76.8ns
Noise reduction	0.58ns	148.4ns	0.39ns	99.8ns
Frame difference	0.38ns	97.2ns	0.21ns	53.7ns

streams. Bit streams of 8 to 1024 bits were generated for each input value. A maximal period LFSR corresponding to the length of each stream was used as the pseudo-random number generator. Processing each bit of the streams requires one clock cycle, so for a 256-bit stream the circuit must run for 256 clock cycles.

The images in Fig. 11(b) show the outputs of processing the sample images using a deterministic, software-based implementation of the algorithms in Matlab. We call these the "golden images" since they have no errors in the outputs. Fig. 11(c) and (d) show the output images produced by the independent and correlated stochastic methods when processing the inputs for 256 cycles. Comparing the output images with the golden images, the mean of the output error rates was calculated as follows:

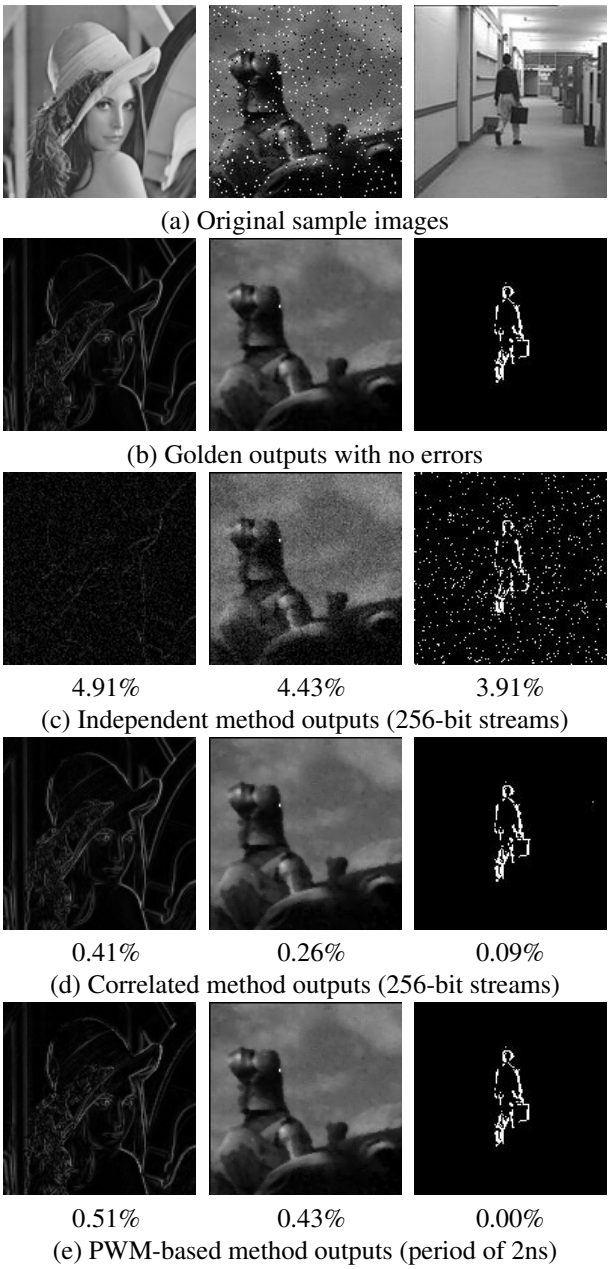
$$E = \frac{\sum_{i=1}^{W}\sum_{j=1}^{H}|T_{i,j} - S_{i,j}|}{255.(W \times H)} \times 100$$

where $S_{i,j}$ is the expected pixel value in the output image, $T_{i,j}$ is the pixel value produced using the circuit, and W and H are the width and height of the image sample. The means of the output error rates for the output images produced by the circuits are shown in Table III. As can be seen in this table, the circuits using correlated bit streams have much better performance than the circuits working on independent input streams.

To evaluate the performance of the low-cost circuits when they are connected to synchronous analog pulses, we implemented the SPICE netlist of the three stochastic circuits. Simulations were carried out using a 45-nm gate library in HSPICE. Image pixel intensities were converted to PWM signals with periods of 0.3ns, 0.5ns, 1ns, and 2ns using the circuit shown in Fig. 12. The implementation cost of the PWM generator, which consists of an analog comparator, a ramp generator, and a clock generator, is a function of its frequency. The area cost

TABLE III
AVERAGE ERROR RATE WHEN PROCESSING THE SAMPLE INPUTS USING THE STOCHASTIC CIRCUITS WHEN THE INPUTS ARE REPRESENTED WITH 1) INDEPENDENT STOCHASTIC BIT STREAMS , AND 2) HIGHLY CORRELATED STOCHASTIC BIT STREAMS.

	Design method	Average Error Rate for different operation time (# of clock cycles)							
		8	16	32	64	128	256	512	1024
Edge detection	Independent	27.5%	19.3%	13.1%	9.02%	6.52%	4.91%	3.70%	2.84%
	Correlated	3.57%	2.41%	1.50%	0.95%	0.62%	0.41%	0.29%	0.20%
Noise reduction	Independent	25.8%	17.3%	11.8%	8.33%	6.06%	4.43%	3.26%	2.42%
	Correlated	6.20%	3.08%	1.59%	0.82%	0.45%	0.26%	0.08%	0.04%
Frame difference	Independent	23.6%	42.0%	30.3%	14.6%	7.53%	3.91%	1.30%	0.48%
	Correlated	80.0%	1.09%	0.16%	0.16%	0.16%	0.09%	0.00%	0.00%



(a) Original sample images

(b) Golden outputs with no errors

(c) Independent method outputs (256-bit streams)

(d) Correlated method outputs (256-bit streams)

(e) PWM-based method outputs (period of 2ns)

Fig. 11. The original sample images and the output images produced by the deterministic implementation of the algorithms and the outputs produced by the independent, correlated, and PWM-based stochastic approaches.

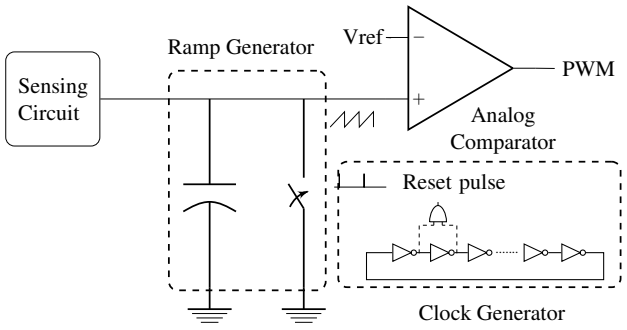


Fig. 12. The design of a low-cost PWM generator. The duty cycle is determined by the current coming from the sensing circuit (a photodiode, a voltage controlled current source, etc.). The Reset pulse defines the frequency of the PWM signal. Vref is a fixed reference voltage

TABLE IV
AVERAGE ERROR RATES OF PROCESSING THE SAMPLE IMAGES WHEN THE INPUTS ARE REPRESENTED USING SYNCHRONIZED PWM SIGNALS.

	Period of PWM input signals			
	0.30ns	0.50ns	1ns	2ns
Edge detection	1.56%	1.02%	0.70%	0.51%
Noise reduction	1.33%	0.91%	0.65%	0.43%
Frame difference	0.02%	0.00%	0.00%	0.00%

of the PWM generator when generating PWM signals with a period of 2ns is roughly as expensive as the area cost of the conventional SN generator [12] with an 8-bit LFSR ($150\mu m^2$). Decreasing the period of the signals lowers the implementation costs since a less expensive clock generator will be sufficient.

We processed the sample images with the low-cost stochastic circuits in HSPICE. Each image pixel was converted to its corresponding PWM signal and processed by the circuit separately. Measuring the fraction of the time the circuit's output signal was high, we computed the output from processing each image pixel. Fig. 11(d) shows the output images when using synchronized PWM signals with a period of 2ns. The means of the output error rates for the produced images are shown in Table IV. As can be seen in this table, the edge detection circuit using PWM signals with a period of 1ns produces an output image with an average error rate of less than 1.0%. This is also the case for the noise reduction circuit using signals with a period of 0.50ns, and for the frame difference circuit using

signals with a period of 0.30ns. Since only one period of the PWM signals is sufficient for determining an accurate output, the circuits' operation time when using the PWM signals is much lower than the latency when processed using the conventional bit-stream representation (see Table.II).

VI. Sources of Computational Errors

There are three primary sources of inaccuracy in performing stochastic operations on synchronized PWM signals:

1. E_G = error in generating PWM signals.

Each PWM generator has some level of inaccuracy in converting the input values into the corresponding PWM signals. This generation error is a function of the period of the signal and the duty cycle and is defined as the difference between the expected and the measured duty cycle. The PWM generator we used in our simulations had an average error rate of 0.23% in converting real numbers in the [0, 1] interval (with 10^{-3} precision) to the corresponding PWM signals with a period of 0.3ns. When generating PWM signals with periods of 0.5ns, 1ns, and 2ns the average error rates were 0.12%, 0.10%, and 0.09%, respectively.

2. E_S = error due to skew between input signals.

Perfectly synchronized PWM signals are necessary to generate accurate outputs in the stochastic circuits that need highly correlated inputs. On-chip variations or other noise sources affecting clock generators can result in deviations from the expected period, phase shift or slew rate of the signals. While these variations can affect the accuracy of the output signal, the results are still accurate to within the error bound expected for stochastic computation.

3. E_M = error in measuring the output signals.

A simple analog integrator can be used to measure the fraction of time the output signal is high. Longer rise and fall times, along with imperfect measurement of the high and low voltages corresponding to digital 1 and 0 values, results in inaccuracies in measuring the correct output value. We compared the output values measured by our SPICE-level implementation of the integrator with the expected values from measuring the outputs under ideal signal levels. The average error rate of the measurements was less than 0.10% for the implemented circuits.

VII. Conclusions

Reducing the area of stochastic circuits is one of the main advantages of exploiting correlation when designing stochastic circuits. In this paper, we proposed a new low-cost sorting unit using only one AND and one OR gate and a novel low-cost SN comparator using a single D-type flip flop. By using the proposed units, we showed how to implement a much less expensive stochastic circuit for two previously implemented algorithms, namely, median filter noise reduction and frame difference-based image segmentation. We introduced synchronous analog pulses as a new way of representing correlated SNs. As a solution to the long latency problem of stochastic circuits, we proposed to use highly overlapped PWM signals. By using these signals not only can we still have the area saving advantage of exploiting correlation in stochastic design, we can also produce highly accurate results after performing the operations for only one period of the input signals. The circuits' operation time is reduced dramatically when working on synchronous analog pulses instead of the conventional stochastic bit-streams.

Acknowledgment

This work was supported in part by National Science Foundation grant no. CCF-1408123. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. The authors would like to thank Marc Riedel, Kia Bazargan, and Ramesh Harjani for helpful discussions, and Shiva Jamali-Zavareh for providing the PWM generator and the analog integrator used in the simulations.

References

[1] A. Alaghi and J. Hayes. Exploiting correlation in stochastic circuit design. In *Computer Design (ICCD), 2013 IEEE 31st International Conference on*, pages 39–46, Oct 2013.

[2] A. Alaghi and J. P. Hayes. Survey of stochastic computing. *ACM Trans. Embed. Comput. Syst.*, 12(2s):92:1–92:19, May 2013.

[3] A. Alaghi, C. Li, and J. Hayes. Stochastic circuits for real-time image-processing applications. In *Design Automation Conference (DAC), 2013 50th ACM / IEEE*, pages 1–6, May 2013.

[4] B. Brown and H. Card. Stochastic neural computation. i. computational elements. *Computers, IEEE Transactions on*, 50(9):891–905, Sep 2001.

[5] B. Gaines. Stochastic computing systems. In J. Tou, editor, *Advances in Information Systems Science*, Advances in Information Systems Science, pages 37–172. Springer US, 1969.

[6] J. Hayes. Introduction to stochastic computing and its challenges. In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*, pages 1–3, June 2015.

[7] B. Li, M. H. Najafi, and D. J. Lilja. Using stochastic computing to reduce the hardware requirements for a restricted boltzmann machine classifier. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '16, pages 36–41, New York, NY, USA, 2016. ACM.

[8] P. Li and D. Lilja. Using stochastic computing to implement digital image processing algorithms. In *Computer Design (ICCD), 2011 IEEE 29th International Conference on*, Oct 2011.

[9] P. Li, D. Lilja, W. Qian, K. Bazargan, and M. Riedel. Computation on stochastic bit streams digital image processing case studies. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 22(3):449–462, March 2014.

[10] M. H. Najafi, D. J. Lilja, M. Riedel, and K. Bazargan. Polysynchronous stochastic circuits. In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2016.

[11] 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*, 24(2):808–812, Feb 2016.

[12] W. Qian, X. Li, M. Riedel, K. Bazargan, and D. Lilja. An architecture for fault-tolerant computation with stochastic logic. *Computers, IEEE Transactions on*, 60(1):93–105, Jan 2011.