An Overview of Time-Based Computing with Stochastic Constructs

Computing on time-based data is a recent evolution of research in stochastic computing (SC). As with SC, complex functions can be computed with low area cost, but the latency and energy efficiency are favorable compared to computations on conventional binary radix. This article reviews the design and implementation of arithmetic operations on time-encoded signals and discusses the advantages, challenges, and potential applications.

M. Hassan Najafi, Shiva Jamali-Zavareh, David J. Lilja, Marc D. Riedel, Kia Bazargan, Ramesh Harjani
University of Minnesota, Minneapolis

Stochastic computing (SC), a paradigm first introduced by W.J. Poppelbaum[1] and Brian Gaines[2] in the 1960s, has received considerable attention in recent years, particularly after Weikang Qian and colleagues reintroduced the concept to the electronic design automation community.[3,4] It has since been explored as a potential paradigm for emerging technologies and "post-CMOS" computing. SC systems have very low area cost. This generally translates to low power consumption, making the paradigm interesting for ultra-low-power processing systems.

In SC systems, logical computation is performed on random bitstreams called *stochastic numbers* (SNs). Two representations are used:

- 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 1 and probability $1 - x$ of being 0.
- In the *bipolar* representation ($-1 \leq x \leq 1$), each bit in the stream has a probability $(x + 1)/2$ of being 1 and $1 - (x + 1)/2$ of being 0.

For example, 10011, 10101, and 11100 are all SNs representing 0.60 in the unipolar and 0.2 in the bipolar representations.

SC offers some intriguing advantages over conventional binary radix. Complex functions can be implemented with simple hardware. This enables the design of low-area and

Published by the IEEE Computer Society
0272-1732/17/$33.00

low-power arithmetic units. For instance, multiplication can be performed with a single AND gate, and scaled addition can be formed with a single multiplexer unit. Also, SC provides tolerance to soft errors (that is, bit flips),[4] timing errors,[5] and clock skew.[6] The obvious disadvantage of SC is the latency. A stochastic representation is exponentially longer than conventional binary radix. This translates to long operation times, particularly if high accuracy is required.[7] Long bitstreams can be compensated for, to some extent, by shortened clock cycles. Nevertheless, long latencies translate into high energy consumption and so offset any gains made by simplified hardware.

This article explores an evolution of the concept of SC. Instead of encoding data in space, as random bitstreams, we encode values in time. The time encoding consists of periodic signals, with the value encoded as the fraction of the time that the signal is in the high (on) state compared to the low (off) state in each cycle. We call these *pulse-width modulated* (PWM) signals (see Figure 1).

Our approach is motivated by the observation that, as technology has scaled and device sizes have gotten smaller, the supply voltages have dropped while the device speeds have improved.[8] Control of the dynamic range in the voltage domain is limited; however, control of the length of pulses in the time domain can be precise.[8,9] Encoding data in the time domain may be more accurate and efficient than converting signals into binary radix.

This time-based representation is an excellent fit for low-power applications that include time-based sensors, such as image processing circuits in vision chips. Converting a variety of signals from an external voltage to a time-based representation can be done much more efficiently than a full conversion to binary radix. This enables a savings of at least 10 times in power at the outset.[10]

By exploiting pulse width modulation, 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 same logical structures used in 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.[6] For example, if a signal is high 68.7 percent of the time, it is evaluated as 0.687 (see Figure 1).

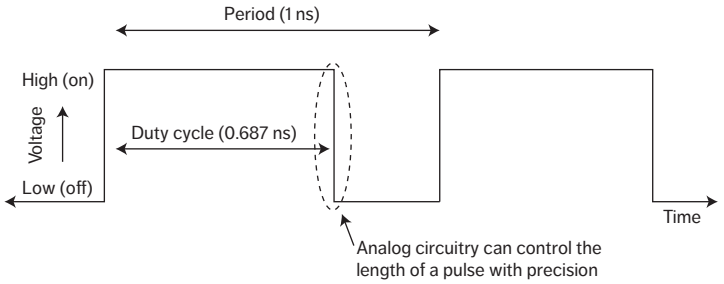

Figure 1. Encoding in time with a periodic analog signal. The value represented is the fraction of the time that the signal is high in each cycle—in this case, 0.687.

This article reviews a transformative new idea: a technique for performing computation on time-encoded analog values directly with ordinary CMOS digital logic.[10] This is related to work on a deterministic approach to SC.[10–12] We have shown that, if properly structured, computation on deterministic bitstreams can be performed with the same circuits as are used in SC, yielding the following benefits:

- Unlike stochastic methods, our deterministic methods produce completely accurate results, not approximations, with no errors or fluctuations.
- The cost of generating deterministic streams is a small fraction of the cost of generating streams from random or pseudorandom sources.
- Most importantly, the latency is reduced by a factor of $1/2^n$, where n is the equivalent number of bits of precision in binary.

Computation on signals encoded in time is directly analogous to this deterministic approach to SC. In this article, we review the performance of different stochastic operations for data processing of inputs generated by a sensing circuit; such data is time-encoded with PWM signals. We discuss the advantages, challenges, and potential applications for computation on such time-encoded signals.

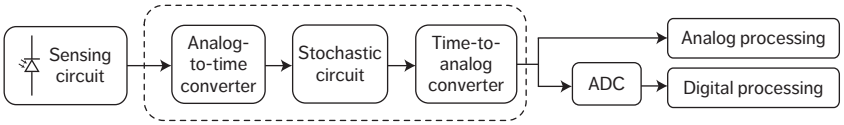

Figure 2. Time-based computing with stochastic constructs. An ATC converts the sensed data to a time-encoded pulse signal. The converted signal is processed using the stochastic circuit, and the output is converted back to a desired analog format using a TAC.

Time-Based Encoding of Stochastic Numbers

Conventionally, the inputs to stochastic circuits are random bitstreams. Sensing circuits, such as image sensors, convert the sensed data (for example, light intensity) to an analog voltage or current. The voltages or currents are then converted to digital form, as binary radix, with costly analog-to-digital convertors (ADCs). Finally, stochastic bitstream generators, consisting of random number generators (that is, linear-feedback shift registers) and comparators, are used to convert the data from binary radix format to stochastic bitstreams.[4]

Recent work has demonstrated low-cost converters that directly convert sensed data from analog form to stochastic bitstreams.[13,14] These greatly reduce the hardware footprint and power consumption of the front end of stochastic circuits. Nevertheless, due to the long latency of operating on random bitstreams, the overall energy consumption—defined as the integral of power consumption over time—remains high. In particular, when high accuracy is needed, the length of stochastic bitstreams becomes prohibitive (for example, more than 1,024 cycles). Even with a higher working frequency, the latency is high; this makes stochastic processing of digital bitstreams inefficient in terms of energy.

However, with sensors that produce time-encoded outputs, which in turn become inputs to the SC circuit, we can work directly with these analog signals instead of converting them into digital bitstreams. This results in a significant saving in energy at the front end. Another compelling advantage is the improvement in the processing time. By using time-encoded signals, the total processing time can be reduced to a time equal to only one clock cycle.[12] The precision of the computation now depends on the precision of the PWM signal in time, rather than the length of the bitstream. Experimental results on image processing applications show up to 99 percent speedup in performance and 98 percent saving in energy dissipation when processing time-encoded signals instead of conventional digital bitstreams[10,12]

Figure 2 shows the flow of computing on time-encoding signals. Assuming that the sensing circuit's output is in voltage or current form, an analog-to-time converter (ATC) circuit (that is, a PWM signal generator) is used to convert the sensed data to a time-encoded pulse signal. This circuit is very low cost, both in terms of hardware area and energy consumption (approximately 30 μm^2 and 0.08 pJ, respectively, for 1 GHz frequency, when supplying the converter with an external clock source). The converted signal is processed using the same circuit constructs as are used in SC. The output is converted back to a desired analog format using a time-to-analog converter (TAC). This is simply a voltage integrator.

The implementation cost of an ATC, which consists of an analog comparator, a ramp generator, and a clock generator, is a function of its frequency. Increasing the frequency (and thus decreasing the period of the PWM signal) increases the implementation cost of the comparator and ramp generator, but lowers the cost of the clock generator (for example, a lower number of inverters in a ring oscillator leads to a higher oscillation frequency). For frequency ranges of lower than 3 GHz, the clock generator has the dominant cost and so increasing the frequency lowers the total implementation cost of the ATC. However, care must be taken because increasing the frequency lowers the effective number of bit (ENOB) of time-based representation, which might then decrease the accuracy of the computation. For comparable accuracy levels, the synthesis results in our previous work show a 40 percent hardware cost reduction when replacing the conventional SN generator with ATCs in image-processing applications.[10]

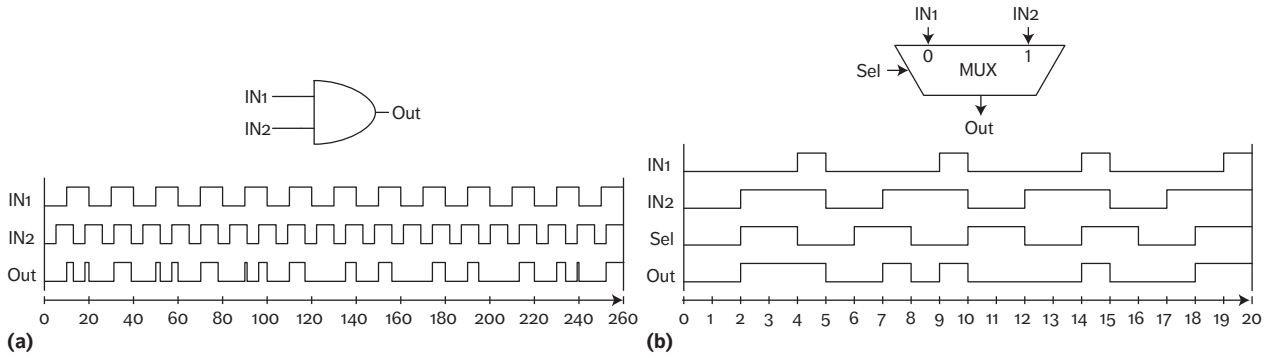

Figure 3. Examples of stochastic operations with independent time-encoded inputs. (a) Multiplying two time-encoded pulse-width modulated (PWM) signals using an AND gate. IN1 represents 0.5 with a period of 20 ns, and IN2 represents 0.6 with a period of 13 ns. The output signal represents 0.30 (78 ns/260 ns), the expected value from multiplication of the inputs. (b) Scaled addition using a multiplexer (MUX). IN1 and IN2 represent 0.2 and 0.6 with a period of 5 ns, and Sel represents 0.5 with a period of 4 ns. The output signal represents 0.40 (8 ns / 20 ns), the expected value from the scaled addition of the inputs.

Independence in Stochastic Circuits

Stochastic operations can be divided into two main categories with respect to correlation between their inputs: operations that require independent (that is, uncorrelated) inputs, and operations that require highly correlated inputs. Multiplication and scaled addition and subtraction are the most common stochastic operations that require independent inputs for correct functionality. An AND gate multiplies two unipolar SNs only if its inputs are independent bitstreams. A multiplexer (MUX) connected to two SNs as the main inputs and another SN as the select input accurately performs scaled addition and subtraction only if the select input is independent of the two main inputs. (Note, however, that the main inputs need not be independent of each other.)

With time-encoded PWM signals, we set the duty cycle to be the value represented. For operations that require independent inputs, such as multiplication using an AND gate or scaled addition using a MUX, PWM signals that are not harmonically related must be used.[10] To see why, consider connecting two PWM signals with the same duty cycle and the same frequency to the inputs of an AND gate. This produces an output equal to the inputs and not the product of the values. Inharmonic frequencies are selected for the input signals, and the operation is run for the least-common multiple (LCM) or multiples of the LCM of the period of the input signals, to produce highly accurate results. Figure 3 shows examples of performing multiplication and scaled addition using time-encoded PWM signals.

Three properties are exclusive to the operations with independent time-encoded inputs:

- *Property 1.* Each independent input must have a frequency inharmonic to the frequencies of other independent inputs. A separate clock source is, therefore, required for each independent input.
- *Property 2.* Increasing the number of independent inputs increases the operation time. The period of the output signal and so the operation time equals the product of the periods (1/frequency) of the independent time-encoded inputs. Thus, by increasing the number of independent inputs, the circuit must run for a longer time to produce accurate results.
- *Property 3.* The accuracy of operations is inversely proportional to the frequency of input signals. Although increasing the frequency lowers the operation time, it decreases the ENOB in representing the input values and so the accuracy in the computations.

Compared to conventional bitstream-based SC, time-encoding the inputs can significantly improve the processing time and hardware area and power cost, and so the energy consumption of operations that require independent inputs.

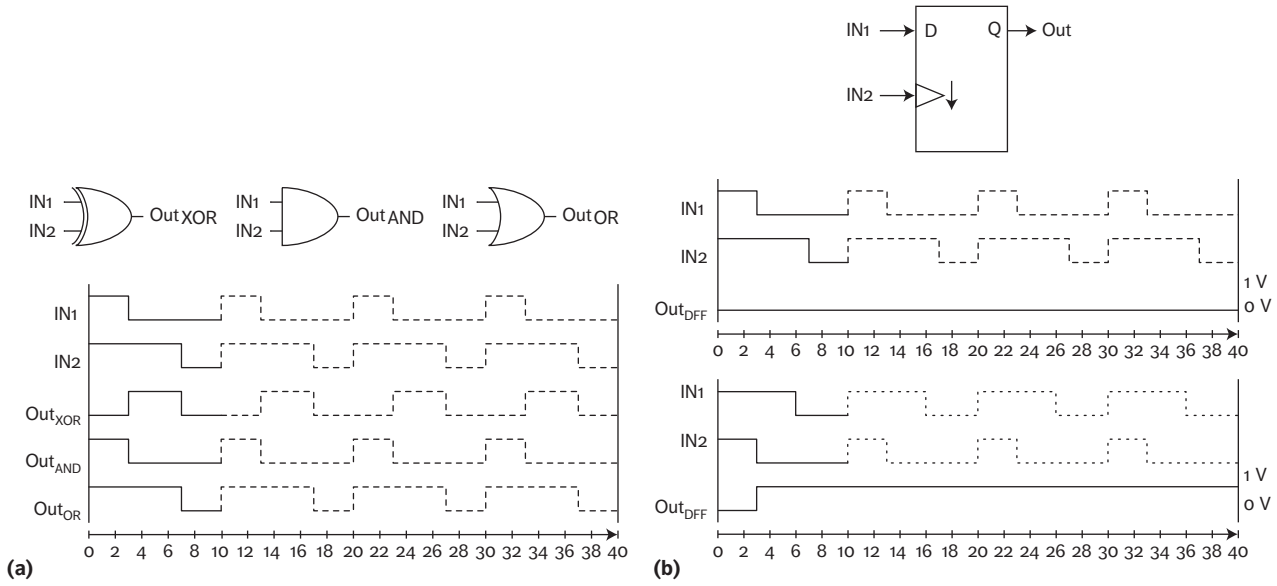

Figure 4. Examples of stochastic operations with correlated time-encoded inputs. (a) Performing stochastic absolute-valued subtraction, minimum, and maximum operations on two synchronized PWM signals: IN1 represents 0.3 and IN2 represents 0.7. Both PWM signals have a period of 10 ns. (b) Comparing stochastic numbers (SNs), represented by synchronized PWM signals, using a D-type flip-flop: (up) IN1 < IN2, and thus Out = 0; (down) IN1 > IN2, and thus Out = 1.

Correlation in Stochastic Circuits

The second category of stochastic operations includes those that require highly correlated inputs. An XOR gate implements absolute-valued subtraction $|x_1 - x_2|$ when it is supplied with highly correlated inputs—that is to say, where the two input streams have maximum overlap in their 1s.[15] 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 for absolute-valued subtraction. This operation is particularly useful in stochastic implementation of image-processing algorithms, such as Robert's cross-edge detection algorithm.[16]

An AND gate with independent inputs works as a multiplier. However, with highly correlated inputs, it gives the *minimum* of the two stochastic streams. An OR gate supplied with highly correlated streams gives the *maximum* of the two stochastic streams. Thus, a basic sorting unit can be constructed with only an AND and an OR gate: supplied with two correlated inputs, it produces the smaller of the two values on one output line, and the greater of the two on the other. Such a low-cost implementation of sorting can save orders of magnitude in hardware resources and power when compared to the costs of a conventional binary implementation. Such circuits are important for applications such as the median filtering noise-reduction algorithm.[17]

Comparison of SNs is another common operation in stochastic circuits. A low-cost stochastic comparator using a simple D-type flip-flop was proposed in our previous work.[12] 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. 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 SNs, N1 and N2, will be 0 if IN1 < IN2, and 1 otherwise.

When representing SNs with time-encoded PWM signals, high correlation or maximum overlap is provided by satisfying two requirements: choosing the same frequency for the signals, and having maximum overlap between the high parts of the signals. For example, two PWM signals that have the same frequency, and each has the high part located

at the beginning or end of each period, are called "correlated" or "synchronized" signals.[12] Figure 4a shows two synchronized PWM signals and the outputs of performing the stochastic absolute-valued subtraction, minimum, and maximum operations on these. Note that the expected output is produced after a single cycle of the PWM input signals. Continuing the operations for additional cycles (the dotted lines) does not improve the accuracy of the results.

Figure 4b also shows two possible cases of comparing SNs, represented by PWM signals using a D-type flip-flop. 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 thus 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.

The exclusive properties of operations with correlated time-encoded inputs include the following:

- *Property 1.* The output of performing stochastic operations on synchronized PWM signals is ready after running the operation for only one period of the input signals. As Figure 4 shows, the fraction of time each output signal is high is the same in all periods of each output signal. In such cases, continuing the operation for additional periods (the dotted lines in the figures) does not change the value or, most importantly, the accuracy of the output.
- *Property 2.* In contrast to stochastic operations with independent inputs that needed time-encoded signals with inharmonic frequencies, the inputs of correlated operations must have the same frequency. Thus, only one source, generating one clock signal, suffices.

Similar to operations that require independent inputs, by time-encoding of inputs, the processing time, area, and power cost, and consequently, energy consumption of operations that require highly correlated inputs can all be greatly reduced when compared to those of the conventional bitstream based processing.

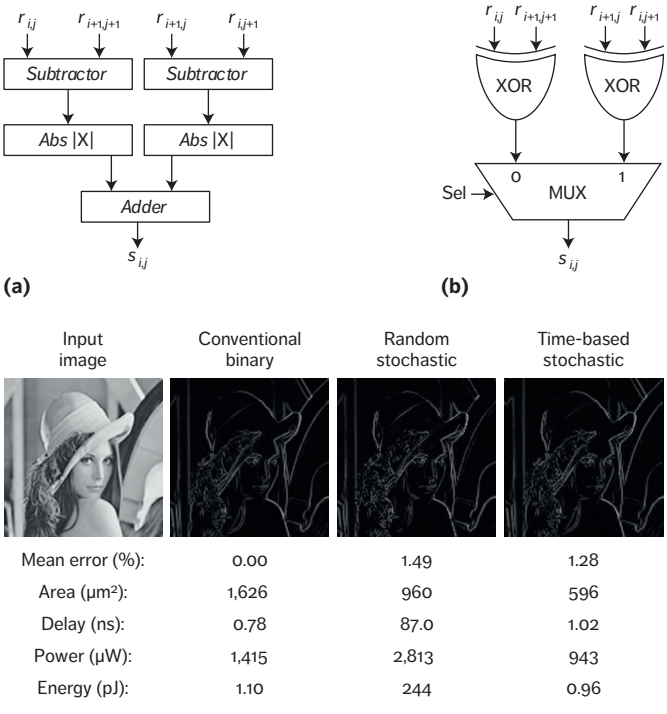

Figure 5. The Robert's cross edge-detection circuit: (a) conventional binary implementation, (b) core stochastic logic, and (c) synthesis results and the results of processing a 128 × 128 sample input image using the binary design, a stochastic design with 256-bit random SNs, and time-based SNs. (For details of the implementations, see our previous work.[10])

Applications

Growth in digital and video imaging cameras, mobile imaging, biomedical imaging, robotics, and optical sensors has spurred demand for low-cost, energy-efficient circuits for image processing. Prior work on SC has shown this computing paradigm's potential in low-cost implementation of image and video-processing algorithms. Image processing based on time-encoded signals could have significant impact in this application area, particularly when power constraints dominate. Time-encoded, mixed-signal processing can be performed on the same chip, with analog-to-time conversion followed by logical computation on the time-encoded signals, using stochastic constructs.

Figure 5 shows the conventional binary implementation and the core stochastic logic for the Robert's cross edge-detection algorithm. The figure summarizes the synthesis results, which are based on a 45-nm gate library. Two sets of numbers are reported: one

for a stochastic design, processing 256-bit random streams, and one for a time-based design. Both of these designs share the same core logic, shown in Figure 5b. The conventional bitstream-based stochastic design uses the random stochastic stream generator proposed by Qian and colleagues.[4] The time-based design uses the ATC proposed in our previous work[10] for time-encoding the inputs.

Considering the critical path of the core stochastic logic as the minimum allowed period of the signals when time-encoding the input data, 0.51 ns is selected as the period of the four main inputs and 0.34 ns is selected for the period of the select input. (For more details on choosing the period of the time-encoded signals, see our previous work.[10]) As Figure 5 shows, the time-based design has significantly lower area and power costs than the conventional binary and stochastic designs. The processing time and the energy consumption are also dramatically improved.

Mixed-signal design is attractive for VLSI implementations of neural networks (NNs) for reasons of speed and energy efficiency. Also, mixed-signal solutions do not suffer from the quantization effects that arise with analog-to-digital conversion. NNs are computationally complex, which makes them a good candidate for processing with low-cost stochastic logic. Digital bitstream-based processing of data in stochastic NN often requires running for more than 1,000 clock cycles to achieve an accuracy close to that of conventional deterministic fixed-point binary designs, which then leads to high energy consumption. Time-based SC has the potential to mitigate these costs, offering energy-efficient designs. Unlike conventional SC, the computations can be completely accurate with no random fluctuation. The approach could have a significant impact in the design of near-sensor NN accelerators.

Challenges

Time-based computing is a mixed-signal technology that combines an analog representation in time with digital processing, using stochastic constructs. In this section, we briefly discuss different challenges in the development and application of method.

Analog Noise

Recent work has shown that by properly structuring digital bitstreams, completely deterministic computation can be performed with stochastic logic.[11] The results are completely accurate with no random fluctuations. Due to the mixed-signal nature of time-based processing, computations on time-encoded signals are susceptible to noise; one cannot promise 100 percent accuracy. Analog noise cannot be completely eliminated from signals and therefore from computation. By careful design of ATC and TAC, and by choosing appropriate frequencies, however, the error can be made very low (less than 0.001 percent mean absolute error).

Resolution

The resolution in time-based processing is limited by noise, rather than by the length of bitstreams, as it is with SC. While there is no limit in the resolution of SNs represented by digital bitstreams, the resolution in our time-encoded approach is limited by the maximum ENOB of the ATC (that is, the PWM generator). For a minimum frequency of 10 MHz, current ATCs can achieve a maximum ENOB of 11 to 12 bits.

Truncation

With time-encoded signals, operations should run for a specific amount of time to produce correct results. For operations with independent inputs, this time equals the product of the period of the input signals; for operations with correlated inputs, it equals the period of the input signals. Running the operation for longer or shorter than the required time results in truncation error.[10] In contrast, stochastic bitstreams have the property of *progressive precision*, meaning that short subsequences of an SN can provide low-precision estimates of its value.[16] The longer the stream runs, the more precise the value. Given enough time, the output converges to the expected correct value, and consequently, the truncation error is generally low.

Synchronization

Operations using synchronized PWM signals are limited to only the first level of logic in a circuit. Providing the required synchronization—that is, having maximal overlap between the

high part of the input signals—is difficult to achieve for the second and higher logic levels.

A naive solution is to convert the output of each level back to an analog format, then perform an analog-to-time conversion and feed this to a higher level. However, this naive method decreases the accuracy and is costly in terms of latency, area, and energy.

Skew

The synchronization must be perfect in operations that require synchronized inputs. On-chip variations or noise sources affecting clock generators can result in deviations from the expected period, phase shift, or slew rate of the signals. Different delays for AND and OR gates, for example, can be a source of significant skew in implementing sorting-based circuits. The skew in each stage is propagated to the next, resulting in a considerable skew error for large circuits. Mitigating the skew by delaying some signals is complex and costly, and may offset gains in area and power.

Rotation

Relatively prime stream lengths, clock division, and rotation were three methods explored by Devon Jenson and Marc Riedel for processing bitstreams deterministically.[11] Choosing inharmonic frequencies for the time-encoded signals corresponds to the "relatively prime" method in Jenson and Riedel.[11] A high-frequency time-encoded PWM signal is connected to the select input of the MUX in previous work by Najafi and Lilja[12] for an accurate scaled addition operation. This approach corresponds to the "clock division" method in Jenson and Riedel.[11] In their "rotation" method,[11] digital bitstreams are stalled for one cycle at powers of the stream length, causing each bit of one bitstream to see each bit of the other stream exactly once. Considering the high working frequency of time-based SC, stalling PWM signals for a very short and precise amount of time might not be possible.

Sequential Circuits

Sequential finite-state machine (FSM)-based approaches exist for implementing complex functions with SC.[18,19] These methods depend on randomness in different ways than combinational methods do. It is not clear how to translate these sequential constructs to deterministic computation on time-based PWM signals.

Computation on time-based encodings offers significant advantages over both deterministic and conventional stochastic approaches. It generally results in circuits that are much less costly in terms of area and power, particularly for applications where the inputs are presented in analog voltage or current form. The savings in the analog-to-time conversion step compared to a full analog-to-digital conversion are significant. Accordingly, the approach is a good fit for low-power real-time image-processing circuits, such as those in vision chips. In future work, we will develop an ultra-low-power video-processing unit using the discussed time-based processing approach. We also use this processing approach in a low-cost, energy-efficient implementation of convolutional NNs and near-sensor NN accelerators

Acknowledgments

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.

References

1. W.J. Poppelbaum, C. Afuso, and J.W. Esch, "Stochastic Computing Elements and Systems," *Proc. Jt. Computer Conf.*, 1967, pp. 635–644.
2. B.R. Gaines, "Stochastic Computing Systems," *Advances in Information Systems Science*, J. Tou, ed., Springer, 1969, pp. 37–172.
3. W. Qian and M. Riedel, "The Synthesis of Robust Polynomial Arithmetic with Stochastic Logic," *Proc. 45th ACM/IEEE Design Automation Conf.*, 2008, pp. 648–653.
4. W. Qian et al., "An Architecture for Fault-Tolerant Computation with Stochastic Logic," *IEEE Trans. Computers*, vol. 60, no. 1, 2011, pp. 93–105.
5. A. Alaghi et al., "Trading Accuracy for Energy in Stochastic Circuit Design," *J. Emerging Technologies in Computing*

Systems, vol. 13, no. 3, 2017, pp. 47:1–47:30.
6. M.H. Najafi et al., "Polysynchronous Stochastic Circuits," *Proc. 21st Asia and South Pacific Design Automation Conf.*, 2016, doi:10.1109/ASPDAC.2016.7428060.
7. J. Hayes, "Introduction to Stochastic Computing and Its Challenges," *Proc. 52nd ACM/EDAC/IEEE Design Automation Conf.*, 2015, pp. 1–3.
8. *International Technology Roadmap for Semiconductors 2.0*, 2015; www.itrs2.net/itrs-reports.html.
9. G.W. Roberts and M. Ali-Bakhshian, "A Brief Introduction to Time-to-Digital and Digital-to-Time Converters," *IEEE Trans. Circuits and System-II*, vol. 57, no. 3, 2010, pp. 153–157.
10. M.H. Najafi et al., "Time-Encoded Values for Highly Efficient Stochastic Circuits," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 5, 2017, pp. 1–14.
11. D. Jenson and M. Riedel, "A Deterministic Approach to Stochastic Computation," *Proc. 35th Int'l Conf. Computer-Aided Design*, 2016, pp. 102:1–102:8.
12. M.H. Najafi and D.J. Lilja, "High-Speed Stochastic Circuits using Synchronous Analog Pulses," *Proc. 22nd Asia and South Pacific Design Automation Conf.*, 2017, pp. 481–487.
13. D. Fick et al., "Mixed-Signal Stochastic Computation Demonstrated in an Image Sensor with Integrated 2D Edge Detection and Noise Filtering," *Proc. IEEE Custom Integrated Circuits Conf.*, 2014, pp. 1–4.
14. N. Onizawa et al., "Analogto-Stochastic Converter using Magnetic Tunnel Junction Devices for Vision Chips," *IEEE Trans. Nanotechnology*, vol. 15, no. 5, 2016, pp. 705–714.
15. A. Alaghi and J. Hayes, "Exploiting Correlation in Stochastic Circuit Design," *Proc. IEEE 31st Int'l Conf. Computer Design*, 2013, pp. 39–46.
16. A. Alaghi, C. Li, and J. Hayes, "Stochastic Circuits for Real-Time Image-Processing Applications," *Proc. 50th ACM/IEEE Design Automation Conf.*, 2013, pp. 1–6.
17. M. Hassan Najafi et al., "Power and Area Efficient Sorting Networks using Unary Processing," to be published in *Proc. IEEE 35th Int'l Conf. Computer Design*, 2017.
18. B.D. Brown and H.C. Card, "Stochastic Neural Computation I: Computational Elements," *IEEE Trans. Computers*, vol. 50, no. 9, 2001, pp. 891–905.
19. M.H. Najafi et al., "A Reconfigurable Architecture with Sequential Logic-Based Stochastic Computing," *ACM J. Emerging Technologies in Computing Systems*, vol. 13, no. 4, 2017, article 57.

M. Hassan Najafi is a PhD candidate and research assistant at ARCTiC Labs in the Department of Electrical and Computer Engineering at the University of Minnesota, Minneapolis. His research interests include stochastic and approximate computing, computer-aided design of integrated circuits, low-power design, and fault-tolerant system design. Najafi received an MSc in computer architecture from the University of Tehran, Iran. Contact him at najaf011@umn.edu.

Shiva Jamali-Zavareh is a PhD student in the Analog Design Laboratory in the Department of Electrical and Computer Engineering at the University of Minnesota, Minneapolis. Her research interests include analog front ends, data converters, RF circuit design, and stochastic computing. Jamali-Zavareh received an MSc in microelectronic circuit design from Aalto University, Finland. Contact her at jamal036@umn.edu.

David J. Lilja is the Schnell Professor of Electrical and Computer Engineering at the University of Minnesota, Minneapolis, where he also serves as a member of the graduate faculties in computer science, scientific computation, and data science. His research interests include computer architecture, high-performance parallel processing, computer systems performance analysis, approximate computing, computing with emerging technologies, and storage systems. Lilja received a PhD in electrical engineering from the University of Illinois at Urbana-Champaign. He is a Fellow of IEEE and the American Association for the Advancement of Science (AAAS). Contact him at lilja@umn.edu.

Marc D. Riedel is an associate professor of electrical and computer engineering with the

University of Minnesota, Minneapolis, where he is a member of the Graduate Faculty of biomedical informatics and computational biology. His research interests include logic synthesis, stochastic computing, and DNA computing. Riedel received a PhD in electrical engineering from Caltech. He has received the Charl H. Wilts Prize for the Best Doctoral Research in Electrical Engineering at Caltech, the Best Paper Award at the Design Automation Conference, and the US National Science Foundation CAREER Award. Contact him at mriedel@umn.edu.

Kia Bazargan is an associate professor in the Department of Electrical and Computer Engineering at the University of Minnesota, Minneapolis. His research interests include stochastic computing, FPGA architectures and applications, and physical design for FPGAs. Bazargan received a PhD in electrical and computer engineering from Northwestern University. He received the US National Science Foundation Career Award. He was an associate editor of *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* and a guest editor of *ACM Transactions on Embedded Computing Systems*. He is a senior member of the IEEE Computer Society. Contact him at kia@umn.edu

Ramesh Harjani is the Edgar F. Johnson Professor with the Department of Electrical and Computer Engineering at the University of Minnesota, Minneapolis. His research interests include analog/RF circuits for wired and wireless communications. Harjani received a PhD in electrical engineering from Carnegie Mellon University. He is the cofounder of Bermai, a start-up developing CMOS chips for wireless multimedia applications. Contact him at harjani@umn.edu.