SHF: Medium: Back to the Future with Printed, Flexible Electronics –
Design in a Post-CMOS Era when Transistor Counts Matter Again

1 Introduction

In the early days of the semiconductor industry, designing complex circuits was an art. Skilled practitioners strove for elegant and efficient designs with the highest performance using the fewest possible transistors. By the early 1970's, with the advent of CMOS integrated circuits, the era of massive scaling had begun. Carver Mead, a semiconductor technology guru at Caltech, exhorted designers to "waste transistors." He coined the term *Moore's Law* based on Gordon Moore's empirical observation that the number of transistors on integrated circuits seemed to be doubling every two years or so. As feature sizes have dropped from millimeters, to micrometers, to nanometers, the transistor counts of integrated circuits have increased from thousands, to millions, to billions. Throughout the history of very large scale integration (VLSI), achievable clock speed and power consumption have been the dominant metrics of the technology. With billions of transistors per integrated circuit, Carver Mead's advice to waste transistors – or at least ignore transistor counts as a metric – seems as prescient as ever.

1.1 Promising Technology that Presents Design Challenges

And yet, CMOS technology is not the only type of technology for making transistors. With remarkable progress in materials science, transistor circuits are being built in all manner of substrates. These include **flexible**, **stretchable**, **conformal** and **impact-resistant** formats [17]. Consider applications such as printed electronics for inexpensive RFID systems. Or distributed sensors in walls or structural materials. Consider an application such as a smart skin on an aircraft wing: pressure sensors detect vortices; actuators respond by deforming the wing in minute ways to counteract these and keep smooth, laminar flow of air across the wing. Or consider an application such as electronics printed on paper that performs image processing functions, for instance photocopying – so paper that can copy paper! Some of these applications are shown in Fig. 1.

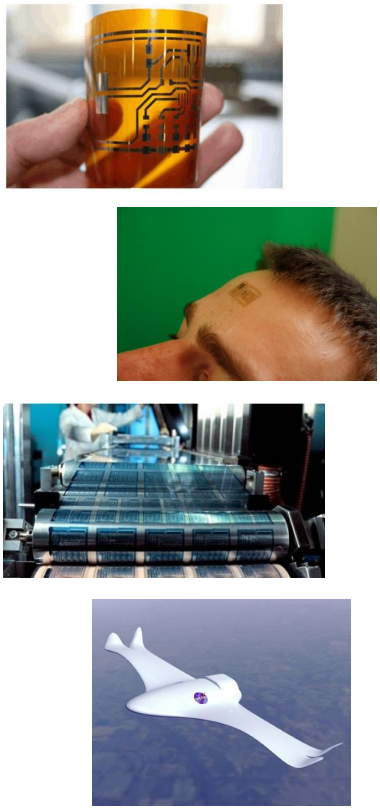

Figure 1: Examples of printed, flexible electronics: RFID tags, biomedical monitoring, display and visual effects, smart wings. Images from [1, 2, 33, 75]

Such technologies have transformed the potential application space for microelectronics, delivering new functional capabilities in radiation detection, health diagnostics, drug-delivery, distributed sensing, information display, food security, identification tagging, inventory tracking, robotics, and human-machine interfacing [18, 21]. A number of approaches to printing electronics are under development world-wide. The PIs have extensive experience in aerosol jet printing, for example, and in this proposal a substantially new, massively parallel and self-aligning process is proposed.

Emerging technologies for flexible electronics have a number of properties that impinge upon design. Often interfacing with conventional CMOS processors is not feasible. For instance, in a smart skin application for an aircraft wing, a huge quantity of local information processing is performed by a large, distributed sensor array. Collecting all the information electronically and transmitting along wires to and from central CPUs would be impractical. The whole point is to build the processing functionality directly into the substrate. Compared to CMOS, flexible electronic systems have **large system-level footprints**, allowing for new opportunities such as ambient energy harvesting (solar, mechanical) from a large area, as well as larger area sensing. On the other hand, the devices themselves have large area because of the limited resolution of current printing technology [21]. Large devices mean that the total device count per unit area is orders of magnitude lower than traditional CMOS. Currently, even 100 printed transistors per cm^2 is difficult to achieve. So, in these applications **transistor counts matter**.

1.2 A Transformative Design Methodology

This proposal seeks to apply the paradigm of **stochastic bit stream computation** to the design challenge of printed electronics. In this paradigm, circuits operate on random bit streams where the signal value is encoded by the probability of obtaining a one versus a zero. With this approach, complex operations

can be performed with very simple logic [56]. For instance, multiplication can be performed with a single AND gate. Because the bit stream representation is uniform, with all bits weighted equally, circuits designed this way are highly tolerant of soft errors (i.e., bit flips).

The paradigm, advocated in prior work by three of the PIs, is a proverbial hammer in want of a nail. The hammer: a method for synthesizing circuits that compute complex functions with remarkably few transistors. The nail: applications where transistor counts matter. The hammer is certainly powerful: compared to conventional design methods, the stochastic paradigm produces designs with significant reduction in transistor counts [43]. It suffers from high latency, and so was never compelling for high-performance, high accuracy computations in CMOS. However, in the context of novel electronic substrates where transistor counts matter, it is a potentially transformative approach. The PIs have published extensively on the topic [39–48, 53–65, 77]. The aim of this project is to provide an **end-to-end validation** of the paradigm of stochastic bit stream computation, applying it to the promising printed, flexible electronics technology. Such an end-to-end system would require improved fabrication methods, developing quality analog to stochastic digital conversion and efficient stochastic processing of *time-varying signals*.

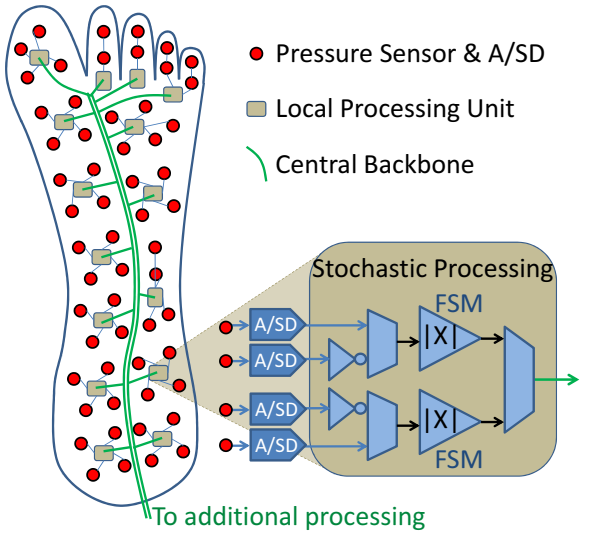

Figure 2: A sensor array with stochastic computational units implemented on a flexible substrate for a robot foot e-skin. A/SD units are Analog to Stochastic Digital converters. Data is locally computed by Stochastic Processing units and the results are sent through a lean communication backbone to a central processing unit.

The proof-of-concept distributed sensing application we propose to develop is shown in Fig. 2. The test platform has a low voltage, flexible pressure sensor array with embedded stochastic computational elements to act as an electronic skin for a robot's foot. Each pressure sensor feeds an Analog-to-Stochastic Digital (A/SD) converter, which provides data to stochastic computational units. These perform local computations such as threshold detection and recognition of spatial and temporal correlations in pressure sensed over an area. The aim is to facilitate a stable walking process as well as to gather data on the terrain (e.g., steepness of slope, sharpness of edges). The platform is discussed in more detail in Sec. 3.2.1.

Broadly, the project will develop the following components to realize the complete, end-to-end system: printing technologies (PI Frisbie); input/output interfacing (PI Harjani); synthesis methodologies (PI Riedel); and architectures and applications (PIs Bazargan and Lilja). Some specific aims are:

- To deliver the proof-of-concept electronic foot skin (Sec. 3.2.1; PIs Bazargan, Frisbie, Lilja, and Riedel).
- To develop elegant A/D and D/A interfaces via sigma-delta modulation to directly convert analog signals to stochastic bit streams and back to analog output signals. We propose significant reductions in the oversampling ratio, which has major ramifications for power consumption of the A/D and D/A circuits (Sec. 3.3; PI Harjani).
- To develop new low temperature, additive manufacturing approaches to printed electronics that will decrease device footprints, thereby simultaneously increasing the device count per area and increasing bandwidth. The goal then will be to manufacture a sophisticated distributed sensor network that highlights the advantages of the new computing architecture with improvements in additive manufacturing of flexible electronic systems (Sec. 3.2.2; PI Frisbie).
- To develop alternate encodings to mitigate latency issues in the stochastic paradigm (Sec. 3.5; PI Bazargan).
- To explore stochastic filtering of time-varying signals using feedback and time delay elements (Sec. 3.1; PIs Bazargan and Harjani).

2 Background

We provide background information on both facets of the project: the design facet, namely the paradigm of stochastic bit stream computation (Sec. 2.1); and the technology facet, namely flexible, printed electronics (Sec. 2.2).

2.1 Logical Computation on Stochastic Bit Streams

Humans are accustomed to counting in a positional number system – decimal radix. Nearly all computer systems operate on another positional number system – binary radix. From the standpoint of *representation*, such positional systems are compact: given a radix b, one can represent b^n distinct numbers with n digits. Each choice of the digits $d_i \in \{0, \ldots, b-1\}$, $i = 0, \ldots, n-1$, results in a different number N in $[0, \ldots, b^n - 1]$: $N = \sum_{i=0}^{n-1} b^i d_i$. However, from the standpoint of *computation*, positional systems impose a burden: for each operation such as addition or multiplication, the signal must be "decoded," with each digit weighted according to its position. The result must be "re-encoded" back in positional form. Any student who has designed a binary multiplier in a course on logic design can appreciate all the complexity that goes into wiring up such an operation.

Consider instead digital computation that is based on a *stochastic* representation of data: 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. These bits can either be serial streaming on a single wire or in parallel on a bundle of wires. When serially streaming, the signals are probabilistic in *time*, as illustrated in Fig. 3(a); when in parallel, they are probabilistic in *space*, as illustrated in Fig. 3(b). Throughout this proposal, we frame the discussion in terms of serial bit streams. However, our approach is equally applicable to parallel wire bundles. Indeed, we have advocated this sort of stochastic representation for technologies such as nanowire crossbar arrays [54].

Consider the operation of multiplication implemented conventionally versus stochastically. A conventional design for a 3-bit carry-save multiplier consists of 30 gates. Fig. 4 shows a stochastic multiplier: it consists of but a *single* AND gate. The inputs are two independent input stochastic bit streams A and B. The number represented by the output stochastic bit stream C is

$$\begin{aligned} c &= P(C = 1) = P(A = 1 \text{ and } B = 1) \\ &= P(A = 1)P(B = 1) \\ &= a \cdot b. \end{aligned} \tag{1}$$

The probability of getting a one at the output, $P(C = 1)$, is equal to the probability of simultaneously getting ones at the inputs, namely, $P(A = 1)$ times $P(B = 1)$. So the AND gate multiplies the two values represented by the stochastic bit streams. In the figure, with bit streams of length 8, the values have a resolution of 1/8. Multiplication is simple and efficient in the stochastic representation precisely because the representation is uniform; no decoding and no re-encoding are required to operate on the values.

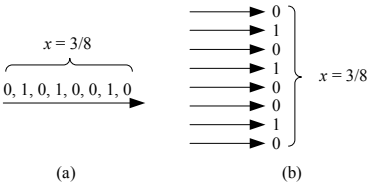

Figure 3: Stochastic representation: (a) A stochastic bit stream; (b) A stochastic wire bundle. A real value x in the unit interval $[0, 1]$ is represented as a bit stream or a bundle. For each bit in the bit stream or the bundle, the probability that it is one is x.

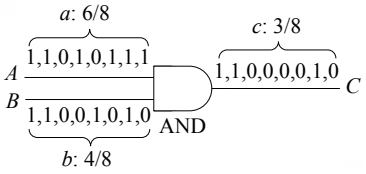

Figure 4: Multiplication with a **stochastic representation**: an AND gate. The inputs are stochastic bit streams A and B and the output is a stochastic bit stream C. Here, the probability of A is 6/8 and that of B is 4/8. The probability of $C = 6/8 \times 4/8 = 3/8$, as expected.

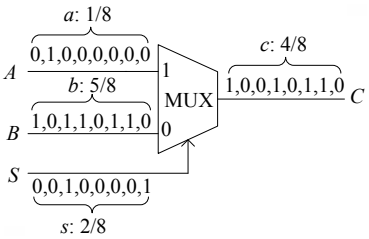

Figure 5: Scaled addition on stochastic bit streams, with a multiplexer (MUX). Here the inputs are $1/8, 5/8$, and $2/8$. The output is $2/8 \times 1/8 + (1 - 2/8) \times 5/8 = 4/8$, as expected.

Consider the operation of addition implemented stochastically. It is not feasible to add two probability values directly; this could result in a value greater than one, which cannot be represented as a probability value. However, we can perform *scaled* addition. Fig. 5 shows a scaled adder operating on real numbers in the stochastic representation. It consists of a multiplexer (MUX), a digital construct that selects one of its

two input values to be the output value, based on a third "selecting" input value. For the multiplexer shown in Fig. 5, S is the selecting input. When $S = 1$, the output $C = A$. Otherwise, when $S = 0$, the output $C = B$. The Boolean function implemented by the multiplexer is $C = (A \wedge S) \vee (B \wedge \neg S)$.

With the assumption that the three input stochastic bit streams A, B, and S are independent, the number represented by the output stochastic bit stream C is

$$\begin{aligned} c &= P(C = 1) \\ &= P(S = 1 \text{ and } A = 1) + P(S = 0 \text{ and } B = 1) \\ &= P(S = 1)P(A = 1) + P(S = 0)P(B = 1) \\ &= s \cdot a + (1 - s) \cdot b. \end{aligned} \tag{2}$$

Thus, with this stochastic representation, the computation performed by a multiplexer is the scaled addition of the two input values a and b, with a scaling factor of s for a and $1 - s$ for b. The discussions above were based on numbers in the range [0,1]. We should note that a linear mapping of the range [0,1] to the range [-1,1] allows us to use stochastic logic operating on probabilities, yet compute on signed numbers [44].

In prior work, we have proposed a general method for synthesizing combinational logic operating on stochastic bit streams [55, 56] (see Fig. 6). We also explored the problem of synthesizing sequential logic, namely finite-state machines, operating on stochastic bit streams [44, 45, 47]. We have also considered the complementary problem of generating probabilistic signals for stochastic computation. We described methods for transforming arbitrary sources of randomness into the requisite probability values, through combinational logic [60], or through sequential logic [64].

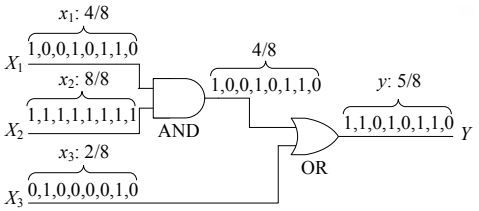

Figure 6: An example of logical computation on stochastic bit streams, implementing the arithmetic function $y = x_1x_2 + x_3 - x_1x_2x_3$. We see that, with inputs $x_1 = 1/2$, $x_2 = 1$ and $x_3 = 1/4$, the output is 5/8, as expected.

Compared to a binary radix representation, a stochastic representation is not very compact. With M bits, a binary radix representation can represent 2^M distinct numbers. To represent real numbers with a resolution of 2^{-M}, i.e., numbers of the form $\frac{a}{2^M}$ for integers a between 0 and 2^M, a stochastic representation requires a stream of 2^M bits. The two representations are at opposite ends of the spectrum: conventional binary radix is a maximally compressed, positional encoding; a stochastic representation is an uncompressed, uniform encoding.

A stochastic representation, although not very compact, has an advantage over binary radix in terms of error tolerance. Suppose that the environment is noisy: bit flips occur and these afflict all the bits with equal probability. With a binary radix representation, in the worst case, the most significant bit gets flipped, resulting in a large error. In contrast, with a stochastic representation, all the bits in the stream have equal weight. A single flip results in a small error. This error tolerance scale to high error rates: multiple bit flips, on average produce small and uniform deviations from the nominal value.

More compelling than the error tolerance is the simplicity of the designs in the stochastic paradigm. Above, we saw how multiplication and addition can be implemented with one and three gates, respectively (three gates to implement the multiplexer for addition). More complex functions such as division, the Taylor expansion of the exponential function, and the square root function can also be implemented with only a dozen or so gates each using the stochastic methodology [55, 56]. Although this is a claim that we can only justify through design examples, we get significant reductions in transistor counts with the stochastic approach. This holds for a wide range of applications, including dedicated circuits for image and signal processing [39, 40, 43].

2.2 Printed Electronics

PI Frisbie has been pursuing research at the forefront of the field of printed electronics. Substantial progress in fabrication and characterization has been made in his laboratory over the last several years. His group has specialized in achieving high performance, low voltage complementary circuits through innovative printing processes.

Transistors and Inverters: Fig. 7 shows a photograph of a printed n-type transistor and the associated transfer and output characteristics. The TFT exhibits excellent characteristics namely an ON/OFF current

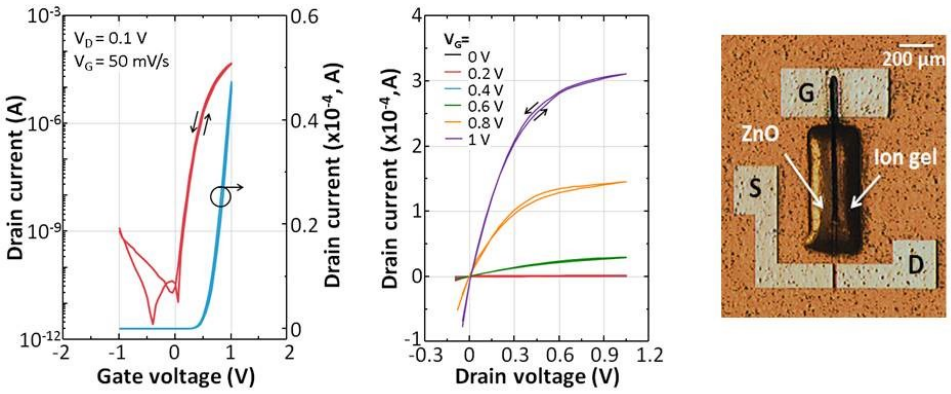

Figure 7: A printed ZnO TFT; transfer and output characteristics.

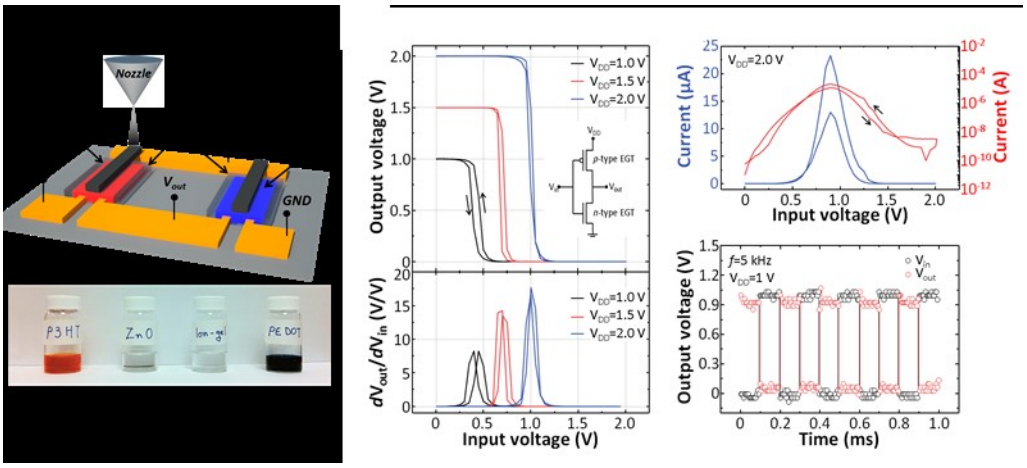

Figure 8: (Left) Scheme of complementary inverter made by aerosol jet printing. Red semiconductor is p-type; blue is n-type. Electronic inks are shown below. (Right) Electrical characterization of the inverter.

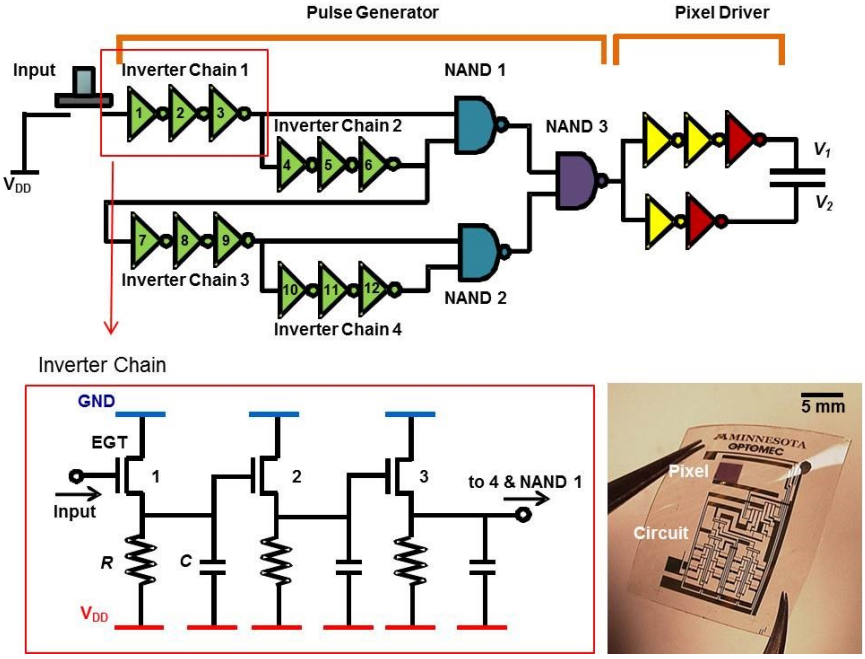

Figure 9: Printed circuit on plastic. The circuit consists of a pulse generator, H-bridge driver and integrated electrochromic pixel. Transistors, capacitors and resistors were all printed by the aerosol jet technique.

ratio of 10^6, electron mobility of 1 cm^2/Vs, threshold voltage (V_T) just positive of 0 V, and a large output current.

Recently, PI Frisbie's group has demonstrated complementary devices prepared by aerosol jet printing, Fig. 8. A polymer semiconductor ink (P3HT) was used to form the p-channel and a ZnO precursor solution was printed to form the n-channel devices. Both the p and n-type transistors have a printed high capacitance gate insulator that allows low operating voltages. The data in the right panel show excellent inverter behavior namely trip voltages at 1 V or below and scaling appropriately with V_{DD}, good device gains above 10, low steady-state power consumption ($<$ 1 nW) and good dynamic switching behavior at 5 kHz. These devices can operate dynamically for many hours without any signs of degradation. Another variant of this inverter structure uses printed carbon nanotubes (CNT) as the semiconductor channel. The CNT inverters operate at 100 kHz, or 10μs stage delays. It is fair to say that for printed electronics, these devices represent the state of the art.

Circuits: Initial steps have also been taken to make prototype circuits. Fig. 9 shows the layout of a pulse delay and H-driver circuit integrated with an electrochromic pixel on a plastic substrate. All components were printed by aerosol jet except the interconnects which were patterned by conventional photolithography. The circuit contains 25 printed p-type TFTs, 22 printed capacitors and 12 printed resistors and operated stably for hours (which was as long as it was tested for).

3 Proposed Work

We discuss, in detail, four specific, synergistic aims of the project: to design and implement stochastic circuitry for image and sensor data processing operations; to design and fabricate a distributed pressure sensor array with embedded stochastic computational enhancement as a step toward "smart electronic skin"; to design and fabricate A/D and D/A interfaces via sigma-delta modulation; to design stochastic filtering for time-varying data; and finally, to explore alternative encodings for lower latency stochastic computation.

3.1 Stochastic Computing Toolkit

Among stochastic circuits we have developed over the past few years, finite-state-machine (FSM) based designs have shown the most promise in terms of area. They are powerful in terms of the range of functions they can approximate, yet they are simple to implement and require remarkably small area. Glossing over many technical details, we present the intuition behind how they work and our future plans for synthesizing complex functions by decomposing them into smaller functions that map to FSMs. We first use the hyperbolic tangent (tanh) function as an example to describe how the FSMs work, and then present an example of functions and constructs one might find in a stochastic computing toolbox.

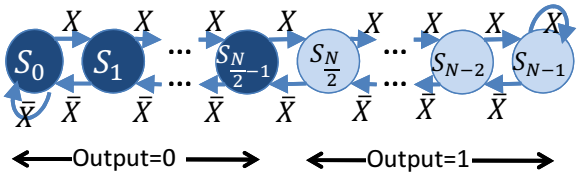

Figure 10: State transition diagram of the FSM implementing the stochastic tanh function.

Fig. 10 shows the state transition diagram for the linear finite-state machine that implements the tanh function, first proposed in [13]. Based upon a stochastic input X, the machine either moves to the next state if $X = 1$ or to the previous state if $X = 0$. The machine stays in S_0 if $X = 0$. It stays in S_{N-1} if $X = 1$. The output is $Y = 0$ (dark-colored states in Fig. 10) when the machine is in states S_0 through $S_{(N/2)-1}$; the output is $Y = 1$ (light color) when the machine is in states $S_{N/2}$ through S_{N-1}. Computing on stochastic bit streams, x is the probability of obtaining a one in the bit stream X and y is the probability of obtaining a one in the bit stream Y. The result is a very good approximation of the tanh function: $y = (e^{\frac{N}{2}x} - e^{-\frac{N}{2}x})/(e^{\frac{N}{2}x} + e^{-\frac{N}{2}x})$.

A simulation of the tanh is shown in the top-right corner of Fig. 11. Intuitively this can be understood as follows. Suppose that the finite-state machine is in a state below the half-way point, one of $\{S_0, \ldots, S_{(N/2)-1}\}$; it is likely to stay below the half-way point, unless it receives far more 1's than 0's. For values of x less than 0.5, it is unlikely to receive far more 1's than 0's. Similarly if it is in a state above the half-way point, one of $\{S_{N/2}, \ldots S_{N-1}\}$, it is likely to stay there unless it receives far fewer 1's than 0's. This is unlikely if x is greater than 0.5. So the circuit performs **thresholding**. Extensive analysis of the linear FSM is given in [43], along with the proof that the structure shown in Fig. 10 indeed approximates the tanh function.

The basis for the analysis is the fact that the FSM construct is a reversible Markov Chain for which we can calculate stationary probabilities, i.e., the probability Π_i that the system is in any given state S_i after it has passed the initial transition period. Given that the stationary probabilities of the states are functions of the input X, we can use an optimization engine to approximate any arbitrary function $f(x)$ by adding a subset of the $\Pi_i(x)$ functions. Furthermore, if we use the state number to choose from a fixed set of constant probability values to route to the output, we can perform weighted addition of the $\Pi_i(x)$ functions, enabling us to approximate a larger set of functions.

Fig. 11 shows an example library of such FSM designs. Many more functions can be implemented using the simple linear FSM construct. Note that the number of flip-flops needed to implement the state machine is exponentially smaller than the number of states. Furthermore, in all our experiments an FSM of size 8 was almost always adequate, rarely requiring a 32-state FSM for high accuracy.

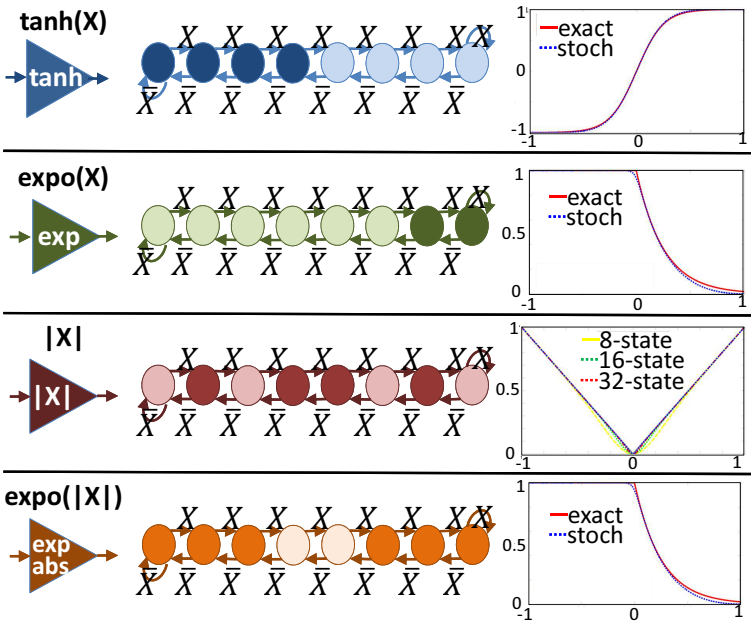

Figure 11: An example FSM library. Each row corresponds to one function. The first column shows the name and the schematic symbol, followed by the state transition diagram (dark states: output=0, light states: output=1) and the plot of the function showing Y as a function of input X. As mentioned in Sec. 2.1, a linear transformation can map the range [-1,1] to probability values [0,1] for stochastic computing.

The FSMs discussed above can be used as "gates" in a library to build more complex functions. Fig. 12 shows a number of examples. The Comparator uses a subtractor (MUX with an inverter input) and thresholding (tanh) logic. The subtractor is similar to the adder in Fig. 5, except that one input is inverted. The combined effect of the subtractor and thresholding is *if* $(P_A < P_B)$ *then* $P_Y \approx 0$; *else* $P_Y \approx 1$, where P_A, P_B, and P_Y are the probabilities of ones in the stochastic bit streams A, B, and Y. The sorter reuses the comparator circuit to output the smaller of the (A, B) pair on the top output line, and the larger one on the bottom. The edge detector unit in Fig. 12 uses a stochastic absolute function to implement Robert's cross operator [24]: $Y_{i,j} = 0.5(|X_{i,j} - X_{i+1,j+1}| + |X_{i+1,j} - X_{i,j+1}|)$. The frame difference hardware uses both the tanh and the absolute functions: its output P_Y is closer to 1 if the difference between the previous frame pixel and the current value is above a constant threshold P_{TH}.

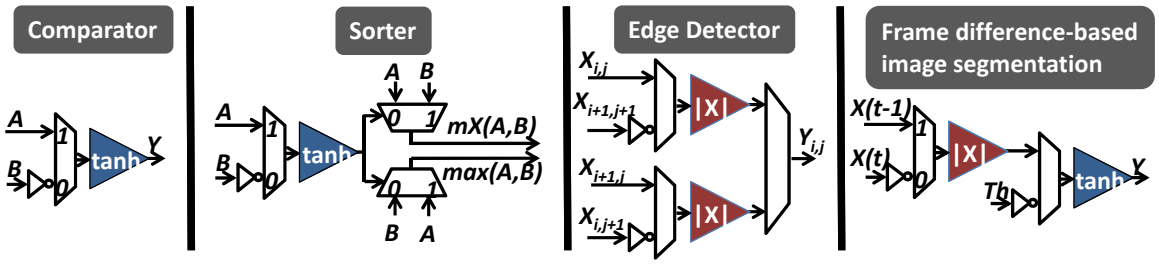

Figure 12: Circuits synthesized using elements of the FSMs library of Fig. 11.

A median filter replaces each pixel with the median of neighboring pixels. It provides excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size [24]. A hardware implementation of a 3×3 median filter based on a sorting network is shown in Fig. 13. The basic unit in this construct, shown by vertical lines, is the sorter of Fig. 12. It is implemented by the stochastic

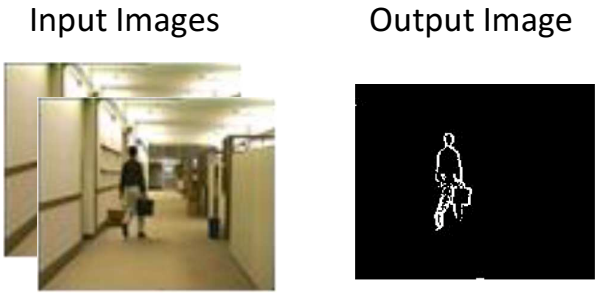

Figure 14: The two frames of input and the output image generated by the stochastic frame difference circuit of Fig. 12.

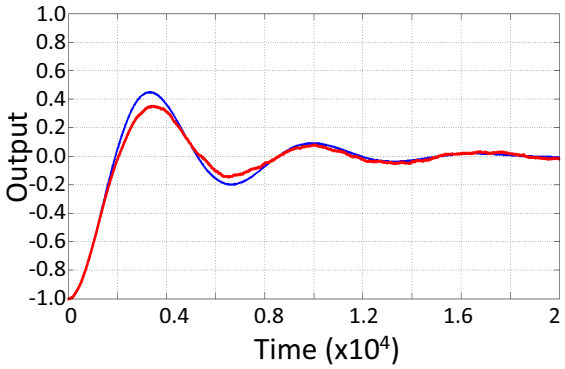

Figure 15: Step response of a stochastic second-order filter (red) compared to that of a real-valued, conventional filter (blue).

comparator described above. Fig. 14 shows sample input images and the corresponding output generated by the stochastic frame difference circuit shown in Fig. 12.

Going forward, our plans for building a more powerful stochastic toolbox include developing stochastic filtering circuits that can work on time-varying signals (e.g., low-pass filter; Sec. 3.2.1). In all our previous published work, we have assumed that input X does not change during the course of the computation. Although that assumption was valid for image processing applications in which pixel values are being read from memory, it no longer holds for applications involving sensor data and time-varying signals of the physical world. The step-response of a preliminary second-order filter designed using two FSMs with a feedback loop is shown in Fig. 15. The blue curve shows a real-valued conventional filter and the red curve shows the output of the stochastic filter. We also plan to devise a synthesis methodology that can break down a large system – such as the median filter discussed above – into smaller components that are compatible with the FSM library. These efforts will be complemented by investigating methods to cut down on latency (Sec. 3.5).

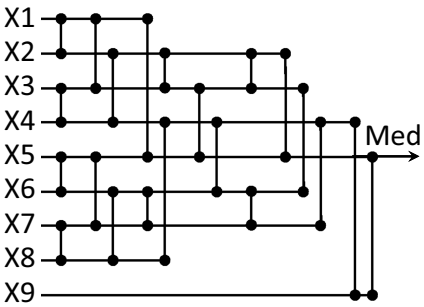

Figure 13: Hardware implementation of the 3×3 median filter based on network of sorting units (each vertical bar is a copy of the circuit "sorter" in Fig. 12).

3.2 Smart Electronic Skin

As a specific example of the advantages of stochastic computing, the PIs will build a low voltage, flexible pressure sensor array with integrated stochastic logic to recognize spatial and temporal pressure correlations. The overall vision and example circuits are discussed in Section 3.2.1, and details of building the sensor array are presented in Section 3.2.2.

3.2.1 Local Stochastic Processing for the Sensitive Robot Foot Skin

As mentioned in Section 1.2, we envision a pressure sensor array with embedded stochastic computational elements fitted to the sole of a robot's foot, as shown in Fig. 2. The foot area is divided into cluster regions containing a number of pressure sensors, each with an Analog to Stochastic Digital converter (Sec. 3.3). Each cluster has a stochastic computational unit that performs local computations, and sends the results through a *lean* communication backbone to a (traditional) processor for further processing and integration into the rest of the system, such as navigation, controlling servo motors and choreographing the movement of differnt parts of the robot. The main advantage of local stochastic computations is savings in the communication backbone, which would be costly in flexible electronics, as well as off-loading part of the computation to the sensor array itself.

The sensor data can be used by the robot either to help it walk more stably or to gather data on the terrain on which it is walking. For a stable walking motion, the robot might need to (1) calculate the slope or the roughness of the surface, (2) determine if it is standing on the edge of a step or a sharp boundary, (3) detect if it is tripping over, (4) verify that it is following a "normal" pattern of pressure changes over time (e.g., the heels show the greatest pressure when they hit the ground first, followed by a wave motion of the pressure to the front of the foot), (5) filter out high-frequency noise generated by vibrations in the ground caused by, e.g., passing vehicles, and (6) sense pressure above a certain threshold on any particular cluster, which might be indicative of stepping on a sharp object with the potential for damaging the e-skin.

The tasks listed above require a number of computational elements, some of which are similar to image processing kernels we have explored in the past, namely edge detection and thresholding, but great challenges remain in implementing more complex functions proposed here – especially when dealing with time-varying signals – and integrating the whole end-to-end system. Edge detection can be used in tasks (1), (2) and (6), and is shown as an example circuit in Fig. 2. Thresholding can be used for task (6). Task (1) would require gradiant calculation. Tasks (3) and (4) are challenging as they require calculating temporal and spatial correlations between sensor outputs. Task (5) requires low-pass filtering.

3.2.2 Flexible Sensor Array

Flexible distributed pressure sensors have been developed by several groups [23, 49, 51, 67, 73] and they are viewed as a first step to developing an "electronic skin" which could be employed in robotics, for example. However, to date the flexible sensors have not been very sensitive to pressure (e.g., > kPa whereas human touch is sensitive to 0.1 Pa) and have required large operating voltages. In addition, they have not incorporated any kind of computational capability that might "pre-analyze" incoming data, such as spatial, temporal or pressure correlations. The ability to incorporate simple computation into distributed sensors will be a powerful approach to enhancing and decentralizing electronic decision making in complex systems like robots. For example, reporting pressure correlations from a distributed pressure sensor e-skin on the foot of a robot may be more useful input to a motion controller than simply an X-Y grid of pressure readings. The point is decentralized information processing from sensors can minimize data overload in a central controller.

A possible single element of the prototype sensor array is shown in Fig. 16, where all components will be prepared by printing/additive processing on plastic or rubber. Low voltage operation will be achieved by using high capacitance gate dielectrics developed in Frisbie's laboratory, and the pressure transducer will be made of a composite, conductive rubber material whose resistance is pressure dependent. Fabrication of the array will be greatly enhanced by the development of new additive manufacturing approaches. That is, in addition to aerosol jet printing approaches, in which Frisbie's group has extensive experience, the PIs will also pursue a scalable, self-aligning strategy to build thin film transistors, pressure transducers, capacitors, resistors and diodes on plastic substrates.

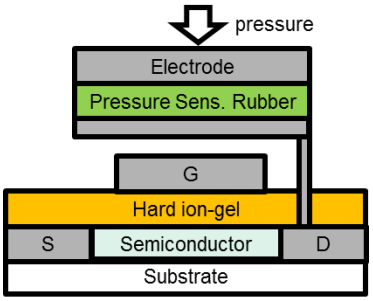

Figure 16: Scheme of a pressure sensor element, which would be part of an array.

This new process is termed Self-Aligned Capillary Flow Lithography (SACFL), and to the PIs knowledge it is a novel approach to printed/flexible electronics. In SACFL, microchannels and reserviors are molded into a coated thermoset material on a plastic substrate by imprint lithography. The dimensions of the channels may range from 100 nm to tens of microns; reservoirs may be hundreds of microns. Electronic inks are delivered to the reservoirs by "drop on demand" dispensers and the liquids, drawn into and along the channels by capillarity, fill relief patterns in the thermoset. The process is self-aligned because multiple inks can be delivered sequentially to the same reservoir, or to different reservoirs, to produce stacked layers of dried materials. The process is also highly parallel because a single reservoir can deliver ink to produce many devices in parallel, and multiple dispensers can be employed to fill multiple reservoirs simultaneously. SACFL is also an additive method because material is only delivered to regions where it needs to be. Related work to our proposed SACFL method include studies on nano-scale capillary flow [27,28], using capillary flow to micro-mold objects [14,36], capillary force lithography [34,74], and using capillaries in the device or final structure [15,32]. To the PIs knowledge, there is currently no established flexible electronics manufacturing

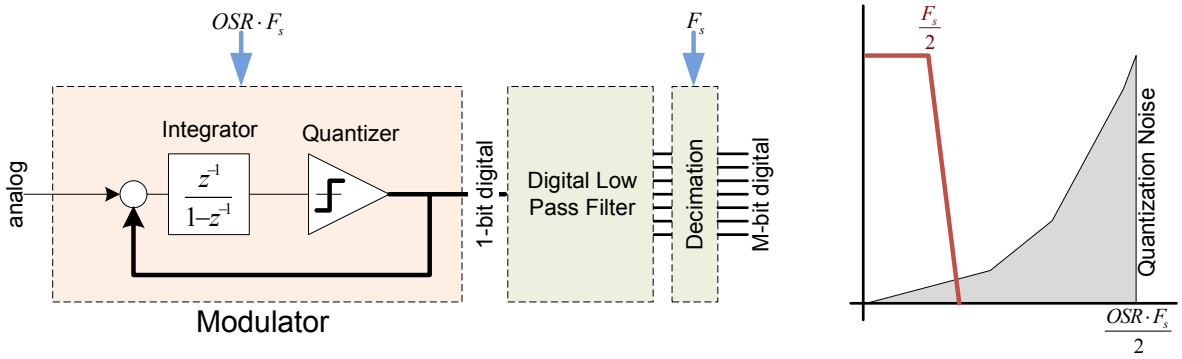

Figure 17: Block diagram for a sigma-delta A/D and the resulting noise shaping and decimation.

process that is simultaneously self-aligning, scalable, additive, and parallel. **Significantly, SACFL solves a central challenge in additive manufacturing of printed electronics – perhaps the central challenge – namely alignment multiple layers of different materials with micron level tolerances (or better!)**. Thus, a key experimental objective of this proposal will be to demonstrate that capillary flow lithography and electronically functional inks may be combined to produce a distributed sensor array as described above.

3.3 A/D and D/A Interfaces via Sigma-Delta Modulation

The digital *stochastic* framework proposed here needs mechanisms to interface to real-world analog signals, i.e., we need circuits to convert analog signals to a stochastic digital signal and vice versa. The obvious first technique that comes to mind is to use a conventional analog-to-digital (A/D) converter followed by a multi-bit digital to single-bit stochastic converter. We propose an alternate, more elegant method that directly converts analog quantities to stochastic digital (A/SD) and stochastic digital to analog (SD/A). The A/SD and SD/A converters proposed here are based on single-bit oversampled converters that are often called sigma-delta ($\Sigma\Delta$) converters [22, 29, 30, 50, 66, 76].

$\Sigma\Delta$ (aka sigma-delta or delta-sigma) converters utilize feedback to introduce noise shaping in the frequency domain such that quantization noise, which is inversely proportional to the conversion resolution, decreases rapidly with increased oversampling (OSR). Fig. 17 shows the block diagram for a 1^{st}-order $\Sigma\Delta$ A/D. In a regular binary digital system, the modulator is usually followed by a low-pass filter and decimation stage. The resulting noise shaping and reduction in the in-band quantization noise is also shown on the RHS of Fig. 17. In Fig. 18 we have plotted the PDF of the output voltage of the integrator, i.e., before the quantizer, for a 2nd order $\Sigma\Delta$ modulator, for an input voltage equal to 0.277350. The comparator basically quantizes this signal over time to give a mean value of 0.277340. We note that this voltage "looks" like a normally distributed stochastic value. In Fig. 19 we plot the FFT of the 1 bit output of the $\Sigma\Delta$ modulator in dB and log(frequency). We note that the quantization noise is moved to the higher frequencies. The flattening from 10^{-4} to 10^{-2} is due the inclusion of KT/C thermal noise in our model. In any real system we are likely to encounter this thermal noise. Additionally, the peak at lower frequencies, near DC, is due to the windowing impact of the DC quantity of 0.277350 at the input.

In our framework we represent values stochastically, i.e., a value of 0.25 is represented by 25% of 1 and 75% of zeros. To represent this value accurately we need to over sample, i.e., operate the 1 bit stochastic value at a higher rate than our original value being represented. This is exactly what a modulator within a $\Sigma\Delta$ converter does. The $\Sigma\Delta$ modulator effectively uses pulse-density modulation, i.e., a value of 0.5 is represented by an equal number of ones and zeros in the time domain. However, it exploits the frequency domain to reduce the oversampling ratio (OSR). This reduction in OSR compared to conventional randomization can significantly reduce the power consumption of the overall system.

We envision a generic stochastic computing system to look as follows: a $\Sigma\Delta$ analog-to-digital modulator, followed by the stochastic engine, which in turn is followed by a $\Sigma\Delta$ digital-to-analog modulator. To ensure that this entire setup is workable we need to ensure that no mathematical manipulation alters the frequency properties of the signal at the final output so that we are able to use a simple $\Sigma\Delta$ digital-to-analog modulator.

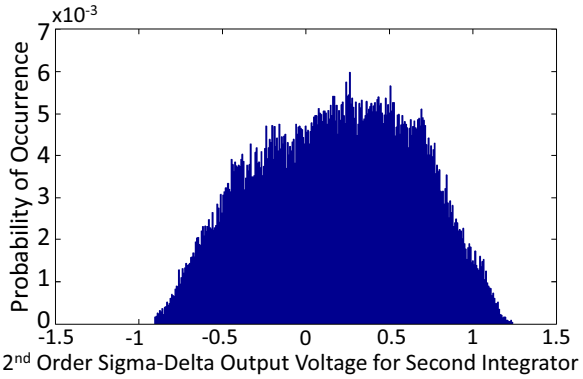

Figure 18: 2nd integrator voltage PDF for 2 stage ΣΔ

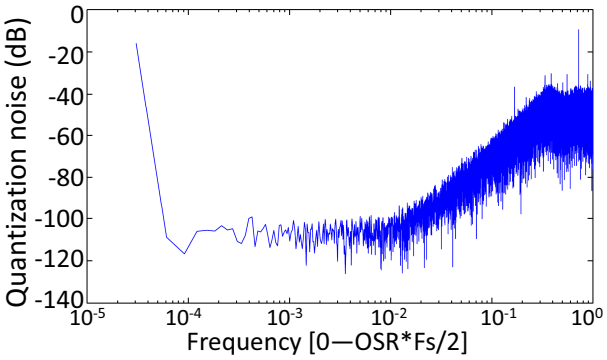

Figure 19: Quantization noise for ΣΔ

In comparison to a traditional ΣΔ converter there is one unique property that needs to be presented by any A/SD converter. When two stochastic values represent the same analog or multi-level digital value, their stochastic representatives need to be uncorrelated. Unlike digital pseudo random generators, analog circuits have "real noise", therefore ΣΔ modulator outputs for the same value are likely to have some variance. However, to ensure that the correlation is small, we may be required to add additional randomness to these converters. There have been many methods proposed over time including dithering and chaos to decorrelate the quantization noise from the input in ΣΔ converters [31,66], including random dither at the input, at the quantizer, partial positive feedback, etc. We will exploit these techniques to develop novel A/SD and SD/A converters that provide the correct translation including the additional decorrelation properties.

Part of the research explorations necessary to develop successful A/SD and SD/A converters include developing low power, low area designs for the ΣΔ analog-to-digital and digital-to-analog modulators, understanding the interplay between the modulators and the mathematical manipulations within the stochastic framework and developing dithering and chaotic techniques that result in stochastic representations of same value being sufficiently uncorrelated.

3.4 Computations with Lowered OSR, the ΣΔ Paradigm

Sigma-delta converters are clearly able to generate single bit stochastic streams for analog quantities using the full oversampling ratio (F-OSR), i.e, representing a 10 bit resolution would require $2^{10} = 1024$ time slots. However, sigma-delta converters perform noise shaping of the quantization noise where the signals are maintained at low frequencies and the noise is pushed to high frequencies. Therefore, a 10 bit resolution can be represented using a 2nd order sigma-delta converter with an oversampling ratio (SD-OSR) of 29, i.e., a reduction of 35× from F-OSR ($2^{10} = 1024$). The savings in oversampling ratio (F-OSR/SD-OSR) increases with resolution requirements and with the order of the sigma-delta converter. However, the complexity of the A/SD and SD/A process increases with order of the sigma-delta. For practical reasons, we shall assume that the order of our implemented sigma-delta module will be limited to two. In Fig. 22 we plot the savings in oversampling ratio (F-OSR/SD-OSR) as a function of N, the binary equivalent resolution for 1st, 2nd and 3rd order ΣΔ converters (blue=1st order, red=2nd order and green=3rd order).

We will be able to exploit the reduced OSR ratio (SD-OSR) of the sigma-delta only if we are able to perform necessary mathematical functions without destroying the noise shaping properties of the sigma-delta. That is to say if we perform a mathematical function between two one-bit stochastic streams that were generated using sigma-deltas the output needs to be a 1 bit stochastic stream and maintain the property that the quantization noise remains at the high frequencies so that it can be filtered out by the ΣΔ digital-to-analog converter at the end. We propose a method to accomplish this using requantization [66]. We illustrate this using an addition example in the next sub-section. It is important to note that in our previous digital stochastic computing work we did not have to consider the frequency domain because we were essentially operating on time-invariant signals (e.g., by assuming the pixel value did not change during the computation).

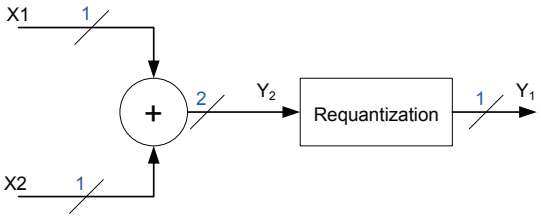

Figure 20: Addition of two 1-bit stochastic streams in the ΣΔ paradigm

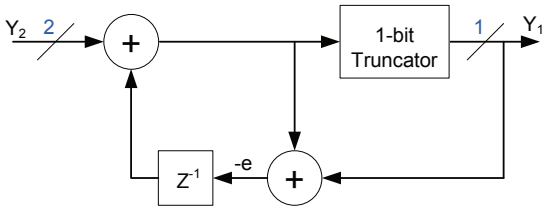

Figure 21: Requantization in the ΣΔ paradigm

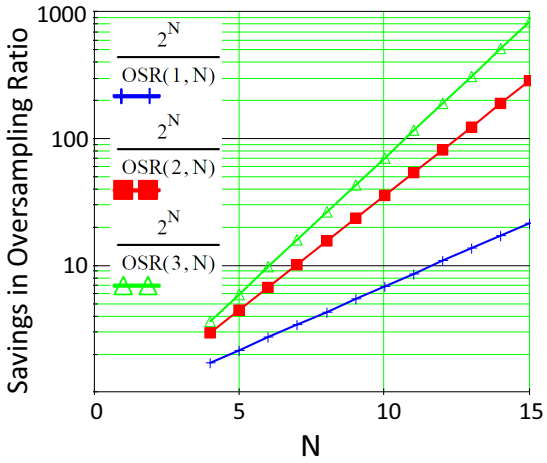

Figure 22: Reduction in oversampling ratio within the ΣΔ paradigm

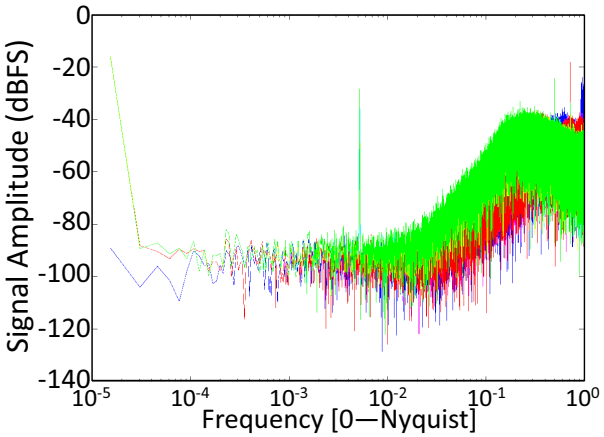

Figure 23: Results of adding two 1-bit stochastic streams in the ΣΔ paradigm

3.4.1 Addition in the ΣΔ Paradigm

This process is illustrated in Fig. 20 and Fig. 21. The two 1-bit stochastic streams are added using a 2 bit adder and the 2-bit output of this adder is requantized to a 1-bit stochastic stream using an all-digital sigma-delta process. The requantization process uses 2 adders (2-3 bits each) and a delay block. The 1-bit truncator looks at the MSB of the value only and does not require additional hardware.

Fig. 23 shows the results of adding two 1-bit streams using the ΣΔ paradigm. X_1, shown in blue is a sinewave with a magnitude of 0.0721 full scale (FS), and a frequency of clock/384, X_2, shown in red, is a DC value 0.2774 full scale and the output, Y, shown in green, is the sum of the two. The X-axis on this plot represents the frequency (log) and the Y-axis represents the value in dBFS. The mean value for X_1 = 3.4E-5 (error due to incomplete cycles of the sine wave plus quantization and thermal noise), the mean value for X_2 is 0.2774 and the mean value for Y is 0.2774 as expected. The DC values show up as skirts of windowing function (Blackman) at the lowest frequency of the plot. We note that the output, Y, contains both the sinewave as well as the DC input, as expected. We note that the low frequency noise in Y is slightly larger than in either X_1 or X_2. This is expected as the noise in the two steams are uncorrelated and add as powers. We also note that the the quantization noise in Y is higher at higher frequencies. This is also not unexpected as we are requantizing the 2-bit value $(X_1 + X_2)$ to a 1-bit quantity, Y, without any additional oversampling. This process will result in increased quantization noise.

We have illustrated the addition function using the ΣΔ paradigm. Part of the proposed research is to identify all the necessary mathematical functions that can be operated in this paradigm and propose simple, low power realizations for each function. Additionally, we will identify design tradeoffs and alternate realization where appropriate.

3.5 Binary Stochastic Number System: Exponential Reduction in Latency

In the introduction, we pointed out two major disadvantages of positional number systems compared to a (unary) stochastic representation: (1) the encoding / decoding overhead and the complexity of operations, and (2) poor fault tolerance, especially when most significant digits are faulty. The major issue causing the complexity of operations stems from costly carry propagation operations to convert the results of addition and multiplication into the canonical representation of the number. Conceptually, issues (1) and (2) are not tightly intertwined and one could come up with alternative encodings that harness the advantages of both unary and positional systems.

We propose a hybrid representation called *binary stochastic* that addresses a major shortcoming of the unary system: its poor representation compactness (linear vs. logarithmic in positional systems). As an example, consider number N representing a real number between 0 and 1 with a resolution of 2^{-7}. To represent this number, we need 7 binary digits in a positional system and $2^7 = 128$ bits to represent it using the unary system. A hybrid encoding can be built using the following encoding: $N = p_1 2^{-1} + p_2 2^{-2} + \cdots p_k 2^{-k}$, where k is the number of binary stochastic digits, and $0 \leq p_i \leq 1$ are *fractional digits* represented using unary bit streams with a resolution of roughly 2^{N-K}. We can use the short hand notation $N = 0.(p_1)(p_2)..(p_K)$ to represent the same number. We will show that as K increases, the length of unary bit streams required for representing the stochastic fractional digits decreases *exponentially*. We should note that binary stochastic changes the range of the numbers to $[0..(1 - 2^{-K})]$, which is a linear scaling of the numbers. To maintain the same accuracy as a unary with 2^N bits, a binary stochastic should have bit stream lengths of:

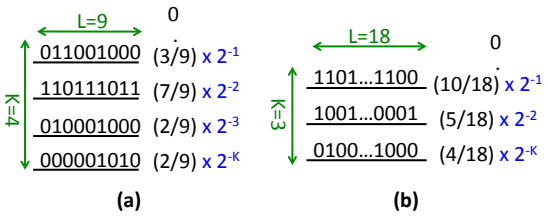

Figure 24: Binary stochastic representation of 0.43 with a resolution of 2^{-7} (equivalent to 128 clocks for unary stochastic) (a) using four binary stochastic wires results in a latency of 9, (b) using three binary stochastic wires results in a latency of 18.

$$L = 2^{N-K}/(1 - 2^{-K}). \tag{3}$$

Even though the binary stochastic system can be considered positional, given that $K \ll N$, the weight difference between the most significant fractional digit and the least significant one is exponentially smaller than that of binary. We can tradeoff the level of fault tolerance vs. representation compactness by changing K.

Fig. 24 shows two binary stochastic examples with different numbers of digits. Compared to unary, which requires a latency of 128 cycles to represent the number 0.43, the binary stochastic examples with K=3 (K=4) require 18 (9) cycles to represent the scaled version of the same number wih the same resolution (128 points within the range), based on Eq. 3.

Since our operations are performed on probabilities, no costly carry propagation operations are required. Scaled addition in binary stochastic translates to K independent scaled additions on bit streams with no carry propagation:

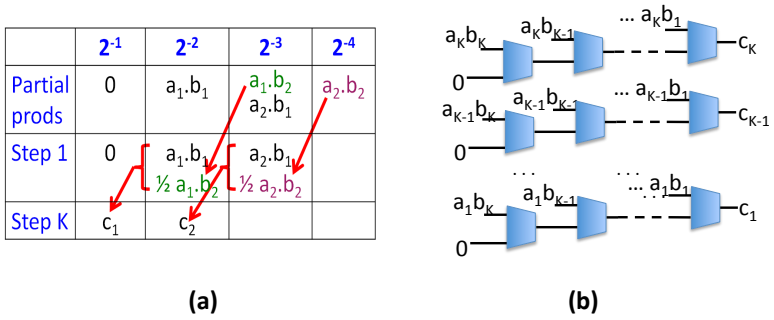

Figure 25: Binary stochastic multiplication: (a) reducing $2K$ bits to K bit outputs (b) the hardware neede for the reduction operation (AND gates to generate partial product bits are not shown.

$$\frac{1}{2}(X+Y) = \frac{1}{2}[x_1 2^{-1} + x_2 2^{-2} + \cdots + y_1 2^{-1} + y_2 2^{-2} + \cdots] = \frac{1}{2}(x_1 + y_1)2^{-1} + \frac{1}{2}(x_2 + y_2)2^{-2} + \cdots \tag{4}$$

where the terms $1/2(x_i + y_i)$ are equivalent to unary scaled addition on digits x_i and y_i.

Multiplication can be done using K chains of MUXes each of depth K, as shown in Fig. 25. The figures show how the example of $0.(c_1)(c_2) = 0.(a_1)(a_2) \times 0.(b_1)(b_2)$ can be calculated. When the first level of

partial products are computed, we get four fractional binary stochastic digits as shown in the first row of the table in the figure. Then a series of reduction operations can be used to pack digits into the two most significant digits to fill the values of c_1 and c_2. Part (b) of the figure shows one hardware realization of the multiplication. The critical path is linear in terms of K (as opposed to K^2 in a conventional binary multiplication).

As discussed in Section 2, our previous published work relied on multiplexers to perform scaled additions, which drop half of the incoming bits and only route the other half to the output. Even though such waste of bits would be acceptable in time invariant values, it would negatively affect the resolution and the range of frequencies of a stochastic bit stream representing a time-varying signal. Our plan is to investigate methods that perform binary stochastic multiplication, possibly with delay elements and accumulators, to overcome this problem (Fig. 20 showed an embodiment of an adder that does this). We have also performed preliminary studies on applying the binary stochastic encoding idea to more complex computations such as Bernstein Polynomials [48]. We have successfully demonstrated exponential reduction in bit stream length with a linear increase in area [81].

Another challenging research question is how to generate time varying independent weighted random fractional digits X_i that represent an input value X. Unlike a binary system where a value has a canonical representation, there are many realizations for the same value X with the K fractional digits X_i (e.g., $0.43 = 0.(0.6)(0.52) = 0.(0.41)(0.9)$). Even though one can use trivial cases such as $X_1 = X_2$, but to realize the most potential binary stochastic can offer in terms of random fluctuations and fault tolerance, we have to study the general case. Our efforts in this regard will be closely coupled to those in Sec. 3.3, A/SD interfacing.

4 Broader Impact

In a debate with an alchemist in 1628, the great French mathematician René Descartes denied the claim that probabilities are as good as certainties in science. Ever since, there has been a lingering stigma associated with estimations and approximations. Those who can, calculate things exactly. Those who can't, simulate and guess. Of course, in many disciplines of science and engineering, probabilistic analysis has become indispensable. However, it is generally applied as a tool for *characterizing uncertainty*: one postulates a definite model and then affixes uncertainties and error margins. In the physical and biological sciences, statistical analysis of data is pervasive. However, such analysis generally is applied as a tool for *inference*: given noisy experimental data, one attempts to extract information that is beyond the reach of direct measurements.

This project advocates stochastic methodologies for *design*. An important goal is to incorporate this viewpoint into the teaching curriculum in electrical and computer computer engineering. Starting with our undergraduate classes – computer engineering, logic design, microcontrollers, and electronics – to our graduate-level classes – VLSI CAD, architecture, analog, and "circuits and biology" – we will teach the students basic probability and develop stochastic concepts such as fault-tolerance, redundancy, and error-correction. Specifically, through these courses, we will we develop the broad theme of computing reliably with unreliable components and computing in terms of statistical distributions.

4.1 Minority Involvement Plan

The PIs will work with the University of Minnesotas College of Science and Engineering Diversity and Outreach program to involve underrepresented students in research. This program manages the NSF-funded North Star STEM Alliance–Minnesotas Louis Stokes Alliance for Minority Participation (LSAMP). One of the core principles of the Diversity and Outreach program is that Mentoring and introduction of research opportunities early in the undergraduate career is the best practice for retention. Through participation in the North Star programs, the students will present their research to North Star fellows to demonstrate their research. They can choose from a selection of outreach events that are provided by the North Star program including a Kickoff Day at the beginning of each year and a spring symposium in the spring semester to showcase research opportunities at the university. Each student will participate in one of these events during their fellowship. The undergraduate students attending these presentations are encouraged by North Star program to seek research positions in labs. North Star also supplies funding for underrepresented students to attend conferences when mentored by a graduate student to increase the exposure of the students to the research community beyond the university's laboratories.

4.2 Undergraduate Involvement in Research

The University of Minnesota offers many research opportunities for undergraduate research. Undergraduate research is supported by the university through the Undergraduate Research Opportunity Program. This is a competitive program that requires the students to write a proposal which gets reviewed and scored. The UROP program funds approximately 80% of the applications providing the students with $1400 stipend and $300 for lab supplies. These students generally are mentored by a graduate student in the lab. This provides graduate students the opportunity to learn mentoring skills and to develop interest in their field. The undergraduates can present their research at the end of the year in an undergraduate research symposium.

4.3 K-12 Outreach Plan

The College of Science Engineering (CSE) offers a summer high school student outreach program, Exploring Careers in Engineering and Physical Science (ECEPS). This program offers students a handson introduction to engineering, science and math opportunities on the University of Minnesota Twin Cities campus by providing the students tours, along with short projects, in different labs around the campus. This program is designed to appeal to and reach both girls and underrepresented minorities with an interest in the STEM disciplines. In particular, two of the four possible one-week sessions are devoted to girls only.

5 Results of Prior NSF Support (One Relevant Grant per PI)

Grant CCF-1241987: "Digital Yet Deliberately Random – Synthesizing Logical Computation on Stochastic Bit Streams"; $299,999; 5/2012 – 4/2014; PIs: Riedel, Bazargan, Harjani, Lilja. **Intellectual Merit**: The concept of Stochastic Computing with state machines to implement combinational logic was introduced. It can significantly reduce hardware cost and allow for trade-offs between accuracy and resource usage. **Broader Impact**: The stochastic synthesis project has resulted in 8 conference papers [39, 40, 45–47, 63–65] and three journal papers [43, 44, 77]. Currently two PhD students are supported by this grant.

Grant CCF-0541162: "MicroStAT: A Microarchitect's Statistically-based Analysis Toolkit" $210,000; 6/2006 – 5/2010, PIs Lilja and Resit Sendag. **Intellectual merit:** The primary intellectual merit of this project was the development of new statistically-based simulation tools for rapidly searching a microprocessor design space. Specific results included a detailed study of the speed and accuracy trade-offs of microarchitectural simulations [38,79], the development of a new metric for determining program phase transitions [37], an evaluation of resampling techniques to compute confidence intervals for harmonic mean-based performance metrics [52], and new approaches for generating and evaluating benchmark subsets [16, 72, 78]. **Broader impacts:** This project developed statistical techniques to assist with database parameter tuning [19, 20]. Also it developed a new complementary branch predictor that achieves high prediction accuracies with low area and power requirements [68]. It supporte research of one Ph.D. student (J. Yi), while providing research opportunities for several others.

Grant 0845650, CAREER Award: "Computing with Things Small, Wet, and Random – Design Automation for Digital Computation with Nanoscale Technologies and Biological Processes"; $500,000; 9/2009 –8/2014; PI Riedel. **Intellectual merit:** This award have established novel and transformative approaches to design automation guided by physical views of computation. A broad theme is the application of expertise from an established field, digital circuit design, to new fields, such as nanotechnology and synthetic biology. **Broader impacts:** The circuit-design community has unique expertise that can be brought to bear on the challenging computational problems encountered in synthetic biology. Applications in biology, in turn, offer a wealth of interesting problems in modeling and algorithmic development. With its cross-disciplinary emphasis, this project will bring new perspectives to both fields. The results have been published in [3–10, 35, 54, 55, 58, 59, 61, 69–71].

Grant ECCS-0925312: "A Sub-2V Printed Flexible Organic RFID System Design for Long Range Communication"; $350,000; 10/2009 - 9/2013; PI Frisbie and Chris Kim. **Intellectual Merit:** Strategies to make low voltage, printed organic transistors and prototype circuits were demonstrated. Also, an organic process design kit was created to facilitate organic circuit design. **Broader Impact:** Two graduate students and one postdoctoral fellow were supported and the award resulted in five publications co-authored with Frisbie [11, 12, 25, 26, 80] with one additional paper currently under review. The organic process design kit was made available to the public online (http://opdk.umn.edu/). Reference [80] received press coverage in the *EE Times* and *MIT Technology Reivew*.

References Cited

[1] Nasa's twist-wing jet explores a radical future. http://www.nasa.gov/missions/research/twist_wing.html.

[2] Liz Ahlberg. Smart skin: Electronics that stick and stretch like a temporary tattoo. http://news.illinois.edu/news/11/0811skin_electronics_JohnRogers.html, 8/11/2011.

[3] M. Altun and M. D. Riedel. Lattice-based computation of Boolean functions. In *Design Automation Conference*, pages 609–612, 2010.

[4] M. Altun and M. D. Riedel. Robust computation through percolation: Synthesizing logic with percolation in nanoscale lattices. *International Journal on Nanoscale and Molecular Computation*, 3(2):12–30, 2011.

[5] M. Altun and M. D. Riedel. Logic synthesis for switching lattices. *IEEE Transactions on Computers*, 61(11):1588–1600, 2012.

[6] M. Altun, M. D. Riedel, and C. Neuhauser. Nanoscale digital computation through percolation. In *Design Automation Conference*, pages 615–616, 2009.

[7] J. Backes and M. D. Riedel. The synthesis of cyclic dependencies with Craig interpolation. In *International Workshop on Logic and Synthesis*, pages 24–30, 2009.

[8] J. Backes and M. D. Riedel. Reduction of interpolants for logic synthesis. In *International Conference on Computer-Aided Design*, 2010.

[9] J. Backes and M. D. Riedel. The synthesis of cyclic dependencies with boolean satisfiability. *ACM Transactions on Design Automation of Electronic Systems*, 17(4), 2012.

[10] J. Backes and M. D. Riedel. Using cubes of non-state variables with property directed reachability. In *Design and Test in Europe*, pages 807–810, 2013.

[11] D Braga, N.C. Erickson, M.J. Renn, R.J. Holmes, and C.D. Frisbie. High-transconductance organic thin-film electrochemical transistors for driving low-voltage red-green-blue active matrix organic light-emitting devices. *Adv. Funct. Mater*, (22):1623–1631, 2012.

[12] D. Braga, M.J. Ha, W. Xie, and C.D. Frisbie. Ultralow contact resistance in electrolyte-gated organic thin film transistors. *Appl. Phys. Lett.*, 97(193311), 2010.

[13] B. Brown and H. Card. Stochastic neural computation I: Computational elements. *IEEE Transactions on Computers*, 50(9):891–905, 2001.

[14] M. Cavallini, C. Albonetti, and F. Biscarini. Nanopatterning soluble multifunctional materials by unconventional wet lithography. *Advanced Materials*, (21):1043–53, 2009.

[15] M. Chabinyc, W. Wong, K. Paul, and R. Street. Fabrication of arrays of organic polymeric thin-film transistors using self-aligned microfluidic channels. *Advanced Materials*, (15):1903–06, 2003.

[16] Vassilios N. Christopoulos, David J. Lilja, Paul R. Schrater, and Apostolos P. Georgopoulos. Independent component analysis and evolutionary algorithms for building representative benchmark subsets. *IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, April 2008.

[17] E. Coatanéa, V. Kantola, J. Kulovesi, L. Lahti, R. Lin, and M. Zavodchikova. Printed electronics, now and future. In *Bit Bang - Rays to the Future*, pages 63–102. Helsinki University Print, 2009.

[18] R. Das and P. Harrop. *Printed, Organic & Flexible Electronics Forecasts, Players & Opportunities*. ID Tech Ex, 2011.

[19] Biplob K. Debnath, David J. Lilja, and Mohamed Mokbel. Sard: A statistical approach for ranking database tuning parameters. *International Workshop on Self-Managing Database Systems (SMDB), IEEE International Conference on Data Engineering (ICDE)*, April 2008.

[20] Biplob K. Debnath, Mohamed F. Mokbel, and David J. Lilja. Exploiting the impact of database system configuration parameters: A design of experiments approach. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Special Issue on Testing and Tuning of Database Systems*, 31(1):3–10, March 2008.

[21] A. Dodabalapur, A.C. Arias, C.D. Frisbie, D. Gamota, T.J. Marks, and C. Wood. *WTEC Panel Report on European Research and Development in Hybrid Flexible Electronics*. WTEC, Baltimore, 2010.

[22] Frank Dropps and Ramesh Harjani. Gain calibration technique for increased resolution in FRC data converters. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 53(11), November 2006.

[23] F. R. Fan, L. Lin, G. Zhu, W. Wu, R. Zhang, and Z. L. Wang. Transparent triboelectric nanogenerators and self-powered pressure sensors based on micropatterned plastic films. *Nano Letters*, 12(6):3109–14, 2012.

[24] R. C. Gonzalez and R. E. Woods. Digital image processing, 3rd edition. *Prentice Hall*, 2008.

[25] MJ Ha, JWT Seo, PL Prabhumirashi, W Zhang, ML Geier, MJ Renn, CH Kim, MC Hersam, and CD Frisbie. Aerosol jet printed, low voltage, electrolyte gated carbon nanotube ring oscillators with sub-5 μs stage delays. *Nano Letters*, (13):954–960, 2013.

[26] M.J. Ha, Y. Xia, A.A. Green, W. Zhang, M.J. Ren, C.H. Kim, M.C. Hersam, and C.D. Frisbie. Printed, sub-3v digital circuits on plastic from aqueous carbon nanotube inks. *ACS Nano*, (4):4388–4395, 2010.

[27] A. Han, G. Mondin, N. Hegelbach, N. de Rooij, and U. Staufer. Filling kinetics of liquids in nanochannels as narrow as 27 nm by capillary force. *J. Colloid Interface Sci.*, (293):151–157, 2006.

[28] J. Haneveld, N.R. Tas, N. Brunets, H.V. Jansen, and M. Elwenspoek. Capillary filling of sub-10 nm nanochannels. *J. Appl. Phys.*, (104):014309, 2008.

[29] Ramesh Harjani. *The Circuits and Filters Handbook*, chapter Integrated Analog-to-Digital Converters. CRC/IEEE Press, 1995.

[30] Ramesh Harjani and Tom Lee. FRC: A method for extending the resolutionof nyquist rate converters using oversampling. *IEEE Transactions on Circuits and Systems II*, pages 482–494, April 1998.

[31] Soren Hein. Exploiting chaos to suppress spurious tone in general double loop sigma-delta modulators. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 40(10), October 1993.

[32] C.E. Hendriks, P.J. Smith, J. Perelaer, A.M.J. Van den Berg, and U.S. Schubert. "invisible" silver tracks produced by combining hot-embossing and inkjet printing. *Advanced Functional Materials*, (18):1031–8, 2008.

[33] ITWissen. Printed electronics. http://www.itwissen.info/definition/lexikon/Tintenstrahldruck-inkjet-print.html.

[34] P.S. Jo, A. Vailionis, Y.M. Park, and A. Salleo. Scalable fabrication of strongly textured organic semiconductor micropatterns by capillary force lithography. *Advanced Materials*, (24):3269–74, 2012.

[35] A. Kharam, H. Jiang, M. D. Riedel, and K. Parhi. Binary counting with chemical reactions. In *Pacific Symposium on Biocomputing*, 2011.

[36] E. Kim, Y. Xia, and G. Whitesides. Micromolding in capillaries: Applications in materials science. *J. Am. Chem. Soc.*, (118):5722–31, 1996.

[37] Sreekumar V. Kodakara, Jinpyo Kim, David J. Lilja, Douglas Hawkins, Wei-Chung Hsu, , and Pen-Chung Yew. Cim: A reliable metric for evaluating program phase classifications. *IEEE Computer Architecture Letters*, 6(1):9–12, January-June 2007.

[38] Sreekumar V. Kodakara, Jinpyo Kim, David J. Lilja, Wei-Chung Hsu, and Pen-Chung Yew. Analysis of statistical sampling in microarchitecture simulation: Metric, methodology and program characterization. *IEEE International Symposium on Workload Characterization (IISWC)*, September 2007.

[39] P. Li and D. J. Lilja. A low power fault-tolerance architecture for the kernel density estimation based image segmentation algorithm. In *International Conference on Application-Specific Systems, Architectures and Processors*, pages 161–168, 2011.

[40] P. Li and D. J. Lilja. Using stochastic computing to implement digital image processing algorithms. In *International Conference on Computer Design*, pages 154–161, 2011.

[41] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. D. Riedel. Case studies of logical computation on stochastic bit streams. In G. Goos, J. Hartmanis, and J.V. Leeuwen, editors, *Lecture Notes in Computer Science: Proceedings of Power and Timing Modeling, Optimization and Simulation Workshop*. Springer, 2012.

[42] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. D. Riedel. Using two-dimensional finite state machine for stochastic computation. In *International Workshop on Logic and Synthesis*, 2012.

[43] P. Li, D. J. Lilja, W. Qian, K. Bazargan, and M. D. Riedel. Computation on stochastic bit streams: Digital image processing case studies. *IEEE Transactions on VLSI Systems*, 2013.

[44] P. Li, D. J. Lilja, W. Qian, M. D. Riedel, and K. Bazargan. Logical computation on stochastic bit streams with linear finite state machines. *IEEE Transactions on Computers*, 2013.

[45] P. Li, D.J. Lilja, W. Qian, K. Bazaragan, and M. D. Riedel. The synthesis of complex arithmetic computation on stochastic bit streams using sequential logic. In *International Conference on Computer-Aided Design*, pages 480–487, 2012.

[46] P. Li, W. Qian, and D. J. Lilja. A stochastic reconfigurable architecture for fault-tolerant computation with sequential logic. In *International Conference on Computer Design*, pages 303–308, 2012.

[47] P. Li, W. Qianand M. D. Riedel, K. Bazargan, and D. J. Lilja. The synthesis of linear finite state machine-based stochastic computational elements. In *Asia and South Pacific Design Automation Conference*, pages 757–762, 2012.

[48] X. Li, W. Qian, M. D. Riedel, K. Bazargan, and D. J. Lilja. A reconfigurable stochastic architecture for highly reliable computing. In *Great Lakes Symposium on VLSI*, pages 315–320, 2009.

[49] S. C. Mannfeld, B. C. Tee, R. M. Stoltenberg, C. V. Chen, S. Barman, B. V. Muir, A. N. Sokolov, C. Reese, and Z. Bao. Highly sensitive flexible pressure sensors with microstructured rubber dielectric layers. *Nature Materials*, (9):859–864, 2010.

[50] Kavita Nair and Ramesh Harjani. A 96dB SFDR 50Ms/s digitally enhanced CMOS pipelined A/D converter. In *IEEE International Solid-State Circuits Conference*, 2004.

[51] Y. Noguchi, T. Sekitani, and T. Someya. Organic-transistor-based flexible pressure sensors using ink-jet-printed electrodes and gate dielectric layers. *Applied Physics Letters*, (89):253507, 2006.

[52] Shruti Patil and David J. Lilja. Using resampling techniques to compute confidence intervals for the harmonic mean of rate-based performance metrics. *IEEE Computer Architecture Letters*, 9(1):1–4, Jan.-June 2010.

[53] W. Qian, J. Backes, and M. D. Riedel. The synthesis of stochastic circuits for nanoscale computation. In *International Workshop on Logic and Synthesis*, pages 176–183, 2007.

[54] W. Qian, J. Backes, and M. D. Riedel. The synthesis of stochastic circuits for nanoscale computation. *International Journal of Nanotechnology and Molecular Computation*, 1(4):39–57, 2010.

[55] 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*, 60(1):93–105, 2011.

[56] W. Qian and M. D. Riedel. The synthesis of robust polynomial arithmetic with stochastic logic. In *Design Automation Conference*, pages 648–653, 2008.

[57] W. Qian and M. D. Riedel. The synthesis of stochastic logic to perform multivariate polynomial arithmetic. In *International Workshop on Logic and Synthesis*, pages 79–86, 2008.

[58] W. Qian and M. D. Riedel. Synthesizing cubes to satisfy a given intersection pattern. In *International Workshop on Logic and Synthesis*, pages 217–224, 2010.

[59] W. Qian and M. D. Riedel. Two-level logic synthesis for probabilistic computation. In *International Workshop on Logic and Synthesis*, pages 95–102, 2010.

[60] W. Qian, M. D. Riedel, K. Barzagan, and D. J. Lilja. The synthesis of combinational logic to generate probabilities. In *International Conference on Computer-Aided Design*, pages 367–374, 2009.

[61] W. Qian, M. D. Riedel, and I. Rosenberg. Uniform approximation and Bernstein polynomials with coefficients in the unit interval. *European Journal of Combinatorics*, 32(3):448–463, 2011.

[62] W. Qian, M. D. Riedel, H. Zhou, and J. Bruck. Transforming probabilities with combinational logic. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (to appear)*, 2011.

[63] W. Qian, C. Wang, P. Li, D.J. Lilja, K. Bazaragan, and M. D. Riedel. An efficient implementation of numerical integration using logical computation on stochastic bit streams. In *International Conference on Computer-Aided Design*, pages 156–162, 2012.

[64] N. Saraf and K. Bazargan. Sequential logic to transform probabilities. In *International Conference on Computer-Aided Design*, 2013.

[65] N. Saraf, K. Bazargan, D. J. Lilja, and M. D. Riedel. Design of sequential logic to generate probabilities. In *International Conference on Computer Design*, 2013.

[66] Richard Schreier and Gabor C. Temes. *Understanding Delta-Sigma Data Converters*. Wiley-IEEE Press, 2004.

[67] G. Schwartz, B. C. K. Tee, J. Mei, A. L. Appleton, D.H. Kim, H. Wang, , and Z. Bao. Flexible polymer transistors with high pressure sensitivity for application in electronic skin and health monitoring. *Nature Communications*, (4):1859, 2013.

[68] Resit Sendag, Joshua J. Yi, Peng fei Chuang, and David J. Lilja. Low power/area branch prediction using complementary branch predictors. *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, April 2008.

[69] P. Senum and M. D. Riedel. Rate-independent biochemical computational modules. In *Proceedings of the Pacific Symposium on Biocomputing*, 2011.

[70] P. Senum and M. D. Riedel. Rate-independent constructs for chemical computation. *PLoS ONE*, 6(6), 2011.

[71] A. Shea, B. Fett, M. D. Riedel, and K. Parhi. Writing and compiling code into biochemistry. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 456–464, 2010.

[72] James Skarie, Biplob K. Debnath, David J. Lilja, and Mohamed Mokbel. Scrap: A statistical approach for creating a compact representational query workload based on performance bottlenecks. *IEEE International Symposium on Workload Characterization (IISWC)*, September 2007.

[73] T. Someya, T. Sekitani, S. Iba, Y. Kato, H. Kawaguchi, and T. Sakurai. A large-area, flexible pressure sensor matrix with organic field-effect transistors for artificial skin applications. *Proceedings of the National Academy of Sciences*, (101):9966–70, 2004.

[74] K. Suh, Y. Kim, and H. Lee. Capillary force lithography. *Advanced Materials*, (13):1386–9, 2001.

[75] Essence Electrostatic Speaker Systems. Essence speakers pioneer printed electronics. http://www.essenceelectrostatic.com/new-essence-electrostatic-speakers-first-to-manufacture-with-printed-electronics/.

[76] Feng Wang and Ramesh Harjani. *Design of Modulators for Oversampled Converters*. Kluwer Academic Publishers, 1998.

[77] W. Xiao, P. Li, and D. J. Lilja. Comparing the performance of stochastic simulation on GPUs and OpenMP. *International Journal of Computational Science and Engineering*, 2013.

[78] Joshua Yi, Resit Sendag, Lieven Eeckhout, Ajay Joshi, David J. Lilja, and Lizy John. Evaluating benchmark subsetting approaches. *IEEE International Symposium on Workload Characterization (IISWC)*, October 2006.

[79] Joshua J. Yi, David J. Lilja, Resit Sendag, Sreekumar Kodakara, and Douglas M. Hawkins. Speed and accuracy trade-offs in microarchitectural simulations. *IEEE Transactions on Computers*, 56(11):1549–1563, November 2007.

[80] W. Zhang, M. Ha, D. Braga, M. Renn, C.D. Frisbie, and C.H. Kim. A 1v printed organic dram cell based on ion-gel gated transistors with a sub-10nw-per-cell refresh power. In *International Solid-State Circuits Conference Digest*, pages 326–328, 2011.

[81] Yanzi Zhu, Peiran Suo, and Kia Bazargan. Binary stochastic implementation of digital logic. In *International Symposium on Field-Programmable Gate Arrays 2014*, submitted.