

On The Theory of Stochastic Processors

Parasara Sridhar Duggirala, Sayan Mitra, Rakesh Kumar and Dean Glazeski

Department of Electrical and Computer Engineering

University of Illinois at Urbana-Champaign

Email: {duggira3,mitras,rakeshk,glazesk1}@illinois.edu

Abstract—Traditional architecture design approaches hide hardware uncertainties from the software stack through overdesign, which is often expensive in terms of power consumption. The recently proposed quantitative alternative of *stochastic computing* requires circuits and processors to be correct only probabilistically and use less power. In this paper, we present the first step towards a theory of stochastic computing. Specifically, a formal model of a device which computes a deterministic function with stochastic delays is presented; the semantics of a stochastic circuit is obtained by composing such devices; finally, a quantitative notion of stochastic correctness, called correctness factor (CF), is introduced. For random data sources, a closed form expression is derived for CF of devices, which shows that there are two probabilities that contribute positively, namely, the probability of being timely with current inputs and the probability of being lucky with past inputs. Finally, we show the characteristic graphs obtained from the analytical expressions for the variation of correctness factor with clock period, for several simple circuits and sources.

I. INTRODUCTION

Moore’s Law states that the number of transistors that can be placed over an integrated chip will double itself for approximately every 2 years. This has been the singular driving force behind the advances in information and communication technologies over the last half-century. However, as transistor sizes decrease exponentially, the resulting effect of manufacturing and environmental uncertainties on the behavior of transistors [1] has become a serious threat to the continuation of Moore’s Law.

Traditional design methodologies completely hide hardware uncertainties from the software stack through overdesign. For example, the operating voltage is often chosen to be at least 20% higher than what is required for correct operation under nominal conditions [2]. Similarly, the frequency of a chip is consistently chosen based on the length of the *slowest* timing path under worst-case conditions. The power and performance costs of such hiding mechanisms are high in present technologies and will become prohibitive in the future as the relative magnitude of uncertainty increases [1]. As a bold alternative, the mantra of the recently proposed *stochastic computing approach* [3], [4], [5] is to relax the notion of correctness and design processors that produce *stochastically correct* results very quickly and efficiently, and rely on hardware and software-based mechanisms for tolerating errors when necessary. For to be an attractive approach, the *rate of errors* should be such that there are system-level power benefits even after accounting for the overhead of tolerating errors.

From the physical standpoint, stochasticity in the outputs comes from the stochastic nature of the stabilization delay that an input data item experiences in the processor. Variations in the the stabilization delay depends on spatio-temporal variations in environmental factors such as temperature, inductive coupling, and voltage noise [6]. Stochasticity may also arise from manufacturing variations that may impact different functional units differently. Finally, aging can cause stochastic variations in delays over relatively longer time scales. While a conventional processor is overdesigned to produce deterministic outputs in spite of these stochastic delays, a stochastic processor allows for stochastic (and hence, possibly wrong) outputs. The error rate dictates the overhead in terms of performance and output quality, and, therefore, a *quantitative analysis* is needed to carefully balance the benefits of hardware stochasticity against the overheads of dealing with errors.

While previous work on stochastic processors has shown considerable power and performance benefits [4], [7], [8] of relaxing correctness, however, a formal framework for design, analysis, testing, and verification of stochastic processors. is missing. Such a framework will enable quantitative analysis of the trade-offs between hardware stochasticity and the overheads of error tolerance, and in its absence, the current architecture and design methodologies [4], [7], [8] are ad hoc at best.

In this paper, we present a first foray into developing a theory of stochastic processing. We begin in Section II, by introducing the model of a *stochastic computing device*—the basic building block of a stochastic processor. A device is the stochastic analogue of a traditional logic gate, however, its output changes *with some delay* after the presentation of the inputs. This delay models the effects of stochasticity in the physical environment of the device and is captured by a (discrete) probability distribution, called *delay distribution*, that has bounded support¹. Because of this unpredictable delay, the observed or latched output at the next observation time (clock cycle) may not be the correct output corresponding to the input. Furthermore, causality of input-output behavior of the devices may prevent an older input that experienced a large delay from producing an output, when a newer input experiences a much smaller delay. Formally, the state of such a stochastic device is essentially a queue that stores all the inputs for which the outputs are yet to be produced, except that more

¹In practice, the delay distributions for different sources of stochasticity can often be obtained from detailed physics-based models and from device and circuit characteristics

recent inputs may obliterate older inputs that experience large delays. Note that this model of stochasticity does not capture the stochasticity caused due to manufacturing variations, but can be easily tuned to capture stochasticity due to aging. Next, we develop the notion of a *stochastic circuit* which is obtained by interconnecting the inputs and outputs of a collection of stochastic devices. We show that the behavior of such circuits fed from random sources of data can be described as a Markov chain.

In Section III, we define what it means for a stochastic circuit to be correct at a given point of observation. This notion is then used to quantify the correctness of circuits in the long run (we call this *correctness factor*), given specific sources of data and specific periodicity of observation. The correctness factor is a key property of a circuit, which influences important design choices including the operational voltage and the clock frequency.

In the Sections III and IV, we show how correctness factors of elementary feed-forward circuits can be computed exactly. The main insight here is the following: an observed output can be correct either because (a) all the current inputs have actually propagated to the output, or because (b) some of the older inputs cause the correct output to appear nevertheless. The probability of (a) is directly obtained from the delay distributions of the relevant devices, and it turns out that the probability of (b) can be analyzed using a key property of a circuit which we call *random correctness probability (RCP)*. The RCP of a circuit with M inputs tells us the likelihood of observing two identical outputs when presented with two sets of random M -bit inputs, of which some subset of input bits overlap. Combining the above we obtain expressions for correctness factors for elementary circuits. In Section V, we present the characteristic graphs for correctness factors of several simple circuits obtained by plugging-in different types of delay distributions and sources, and varying the clock period, in the above-mentioned analytical expressions. These results have also been validated separately through probabilistic model-checking and Monte Carlo simulations (which we do not report here). These graphs corroborate our informal understanding of how correctness of different stochastic computing elements change with frequency scaling, based on detailed architecture-level simulations. The foundations laid out in this paper, we believe, will aid the analysis of more complex stochastic circuits and processors and also will aid in making design choices for stochastic processors with respect to energy consumption, clock-speed, and correctness factor.

II. STOCHASTIC PROCESSOR MODEL

In this section, we first discuss the underlying physical phenomenon corresponding to stochastic processing and then present the mathematical model of stochastic devices and circuits.

A. Physical processing

Basic computing elements are interconnected by wires to create larger and more complex circuits, devices, and systems. Inputs to and outputs from computing elements are voltage signals on wires. Although these signals are real-valued and change continuously with time, for the sake of convenience of modeling we will work with the usual discrete-time discrete-valued abstraction: signals are $\{0, 1\}$ -valued and they change state instantaneously. To be clear, there is a hardware clock in the system, and the edges of this clock determine when values are latched. The periodicity of this clock shows up as the period of the sources and the observation in our paper.

When an input signal to a computing element changes, the corresponding output appears after a delay which is chosen according to some discrete probability distribution, say γ . This stochastic delay models the time it takes for the element to reach a steady state value after the change in the input, and it depends on complex environmental factors such as inductive coupling of the circuit, temperature, and voltage noise.

For two consecutive changes in input I_1 and I_2 separated by a time interval of Δ , if the delay experienced by I_2 is less than sum of Δ and the delay experienced by I_1 , then the output corresponding to I_1 never appears at the output. This preserves causality of the input-output relationship of devices. This property also makes our model for stochastic devices different from standard queuing models.

B. Stochastic Devices as Probabilistic Automata

The set of boolean values $\{0, 1\}$ are represented as \mathbb{B} . For a natural number $N \in \mathbb{N}$, we denote the set $\{1, 2, \dots, N\}$ by $[N]$. For a set S , we denote the set of probability distributions over S by $\mathcal{D}(S)$. For a probability distribution $\gamma \in \mathcal{D}(\mathbb{N})$ over natural numbers, $\gamma(i)$ denotes the probability of choosing $i \in \mathbb{N}$, $\text{supp}(\gamma) \triangleq \{i \in \mathbb{N} \mid \gamma(i) > 0\}$, i.e., $\text{supp}(\gamma)$ is the set of values v for which the $\gamma(v) \neq 0$ and if γ has finite support then $\max(\gamma) \triangleq \max(\text{supp}(\gamma))$. For the cumulative distribution function (CDF) Γ of γ is defined as, $\Gamma(i) \triangleq \sum_{j=0}^i \gamma(j)$ and $\Gamma(> i) \triangleq 1 - \Gamma(i)$.

A *queue* of type T is a sequence of elements of type T . The first element of a queue q is denoted by $q.head$, and the rest of the elements constitute another queue, which we denote by $q.tail$. The total number of elements in a queue is denoted by $q.length$. A *timed queue* of type T is a sequence of elements of type $T \times \mathbb{N}$. The first component of such an element w is called the *value*, denoted by $w.value$, and the second component, $w.deadline$, is called the *deadline*. Thus, $q.head.value$ denotes the value corresponding to the head element of the timed queue q .

The elementary model of a processing unit is a *device*. Roughly, a device stores input(s) from (possibly multiple) input wires and produces outputs according to a specific function and after a certain amount of delay which is given by delay distribution(s).

Definition 1: An N -bit device \mathcal{D} is specified by (a) a collection $\{\gamma_i\}_{i=1}^N$ of N discrete probability distributions over

\mathbb{N} with finite supports and $\min(\text{supp}(\gamma_i)) > 0$; γ_i is called the *delay distribution for the i^{th} input*, and (b) f is a function $\mathbb{B}^N \rightarrow \mathbb{B}$ called the *device function*.

We denote the number of inputs of a device \mathcal{D} by $N(\mathcal{D})$. The CDF for the distributions γ_i is denoted by Γ_i . The device function defines the single bit output as a function of the N input bits. The output, however, does not change instantaneously when the input changes; the delay distributions define the duration of time after which a change in an input is reflected at the output.

Example 1 Figure 1 illustrates the inputs and outputs of three devices: a 1-bit inverter \mathcal{D}_1 , a 2-bit AND device \mathcal{D}_2 , and a 3-bit device \mathcal{D}_3 . Device \mathcal{D}_1 is completely specified by (a) its delay distribution, for example $\gamma_1(1) = \frac{1}{2}$, $\gamma_1(2) = \frac{1}{4}$, and $\gamma_1(3) = \gamma_1(4) = \frac{1}{8}$, and (b) its device function $f \triangleq (y = -b_1)$. ■

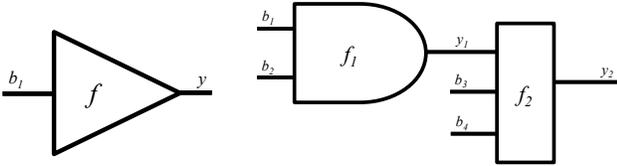


Fig. 1. *Left*: A 1-bit inverter device \mathcal{D}_1 . *Right*: A 2-bit adder circuit with devices \mathcal{D}_2 and \mathcal{D}_3 .

The semantics of a device \mathcal{D} is given in terms of a discrete time-probabilistic state machine, which we also denote by \mathcal{D} (the meaning of the notation will be clear from context). Informally, the state machine \mathcal{D} stores the inputs from all the input wires in respective queues until it is time for the inputs to appear at the output of the device. The state of \mathcal{D} is specified using the following variables: (a) $queue_i, i \in [N]$, a queue of pairs $\langle b, t \rangle$, where $b \in \mathbb{B}$, $t \in \mathbb{N}$; initially each $queue_i$ is empty, (b) boolean valued *head variables* h_i , for each $i \in [N]$, (c) boolean valued *output variable* y . Initially, h_i 's and y are all 0. A state of \mathcal{D} is a valuation of all the above variables. We will denote states of \mathcal{D} by bold letters $\mathbf{x}, \mathbf{x}', \mathbf{x}_1$, etc.

The functions in Figure 2 describe the probabilistic transitions of \mathcal{D} . Specifically, given a state \mathbf{x} , a set of inputs b_1, \dots, b_N , and a set of input delays d_1, \dots, d_N , the next state \mathbf{x}' is obtained by applying *Time*, *Deq*, and the appropriate *Enq* function to \mathbf{x} in sequence. That is, $\mathbf{x} = \text{Enq}(b_1, \dots, b_N, d_1, \dots, d_N) \circ \text{Deq} \circ \text{Time}(\mathbf{x})$. The *Time* function advances time: it decrements the deadlines for each of the previously enqueued inputs that are yet to affect the output. The *Deq* function updates the output and the head variables: first, for any enqueued input $\langle b, t \rangle$ in $queue_i$ for which the deadline t is 0, b is copied to h and removed from $queue_i$. Next, the output y is computed by applying the device function f to the (possibly newly updated) head variables. Finally, the function $\text{Enq}(b_1, \dots, b_N, d_1, \dots, d_N)$ models the arrival of new inputs b_i at the i^{th} wire and experiencing a delay of d_i . This has the effect of wiping out all the past inputs in $queue_i$ that are supposed to experience a delay of d_i or more,

and the addition of $\langle b_i, d_i \rangle$. The transition probability from \mathbf{x} to the above \mathbf{x}' :

$$\begin{aligned} \mu_{\mathbf{x}, \mathbf{x}'} &= \prod_{i=1}^N P[i^{\text{th}} \text{ input} = b_i] P[i^{\text{th}} \text{ input's delay} = d_i] \\ &= \prod_{i=1}^N P[i^{\text{th}} \text{ input} = b_i] \prod_{i=1}^N \gamma_i(d_i). \end{aligned}$$

Thus, at every time instant, if the probability of a certain sequence b_1, \dots, b_N appearing at the input of \mathcal{D} is given, then the transition probability $\mu_{\mathbf{x}, \mathbf{x}'}$ is well-defined and the process \mathcal{D} is a finite state Markov chain.

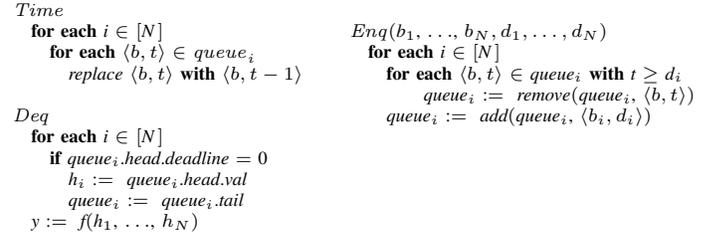


Fig. 2. Transition function for device \mathcal{D} with delay distributions $\gamma_1, \dots, \gamma_N$, and device function f . The choice of the parameters d_1, \dots, d_N in the *Enq* transition are chosen according to $\gamma_1, \dots, \gamma_N$.

Example 2 Consider the device 1-bit \mathcal{D}_1 in Example 1 being fed with a sequence of input bits $0, 1, 0, 1, \dots$. At the beginning of time-step 1, $queue_i$ is empty. At the beginning of step 2, $queue_i$ has a single element $\langle 0, d \rangle$, where $d = 2$ with probability (w.p.) $\frac{1}{2}$, $d = 3$ w.p. $\frac{1}{4}$, and $d = 4$ and $d = 5$, each w.p. $\frac{1}{8}$. The variables h_1 and y continue to remain 0 at this time. ■

C. Stochastic Sources and Circuits

A source in our framework models a source of data encoded as bits which feed into stochastic devices.

Definition 2: For $N, K > 0$ an N -bit K -period source s produces N new binary values every K^{th} instant of time, and the values remain constant for the intervening $(K - 1)$ time periods. The i^{th} bit produced by s at time t is denoted by $s_i(t)$.

Let $t \in \mathbb{N}$. Let t be written as $qK + r$, for some fixed $q, r \in \mathbb{N}, 0 < r < K$, then $s_i(t) = s_i(qK)$. If for every q , $s_i(Kq)$ is the same, then s is a *constant source*; if for every q , $s_i(Kq)$ depends only on $s_i((K - 1)q)$ then s is *Markovian*, more generally, $s_i(Kq)$ may depend on $s_j((K - k)q)$. For this paper, we consider the simple class of *random sources*: For each $i \in [N]$ there is a constant $p_i \in [0, 1]$, such that for any q then

$$s_i(qK) = \begin{cases} 1 & \text{with probability } p_i, \\ 0 & \text{with probability } (1 - p_i), \end{cases}$$

and for any q and r , $s_i(qK + r) = s_i(qK)$. As we shall see shortly, circuits with random sources provide a good starting point for the development of the general theory. Techniques developed for this simpler class, we believe, can be adapted

to other types of sources. We denote the number of outputs and the period of a source s by $N(s)$ and $K(s)$.

A circuit models an interconnected collection of devices and sources such that every input of every device is connected to either an output of some device or a source. For a device \mathcal{D} , we denote its inputs by $\mathcal{D}^1, \dots, \mathcal{D}^{N(\mathcal{D})}$ and its output by \mathcal{D}^y . For a source s , we denote its N outputs by $s_1, \dots, s_{N(s)}$.

Definition 3: A circuit with a collection of \mathcal{D} devices and a collection of \mathcal{S} sources is a function C that maps every output of every source and device to some (possibly empty) set of inputs of devices

$$C : \{\mathcal{D}^y \mid \mathcal{D} \in \mathcal{D}\} \cup \{s_i \mid s \in \mathcal{S}, i \in [N(s)]\} \\ \rightarrow \{\mathcal{D}^i \mid \mathcal{D} \in \mathcal{D}, i \in [N(\mathcal{D})]\}$$

such that, for every $\mathcal{D} \in \mathcal{D}$ and every input \mathcal{D}^i , $|C^{-1}(\mathcal{D}^i)| = 1$. That is, every input is mapped from exactly one output.

Given a circuit C with devices $\mathcal{D}_1, \mathcal{D}_2 \in \mathcal{D}$, \mathcal{D}_1 is said to *precede* \mathcal{D}_2 , if C maps the output of \mathcal{D}_1 to an input of \mathcal{D}_2 . Consider the graph $G_C = (V_C, E_C)$ with the set of vertices $V = \mathcal{D} \cup \mathcal{S}$ and the set of edges $E_C = \{(u, v) \mid u \text{ precedes } v \text{ in } C\}$. If G_C is a DAG then C is said to be a *feed-forward circuit*. A feed-forward circuit consisting of a single device is called a *simple circuit*.

We recursively define the depth of feed-forward circuits. All the sources of a circuit C are at depth 0. A device in C is at depth i , if all its predecessors are in depth $(i-1)$ or less, with at least one predecessor exactly at depth $(i-1)$. A simple circuit has a depth of 1.

Finally, we describe the semantics of feed-forward stochastic circuits (henceforth, simply circuits). A circuit computes a function of the bits produced by the sources by applying a sequence of transformations to these bits through devices. There are two possible interpretations of a circuit: (i) static and (ii) dynamic:

The static interpretation tells us how the circuit behaves in the steady state or in the long-run under fixed inputs. That is, the output observed from the circuit when the sources are fixed to constant bits, and as time goes to infinity. This static or steady-state behavior of a circuit C is captured by the *circuit function* and is denoted by f_C : it assigns values to the output variables of all the devices in C as a function of the inputs from the (constant) sources. The circuit function can be expressed in terms of the device functions recursively as follows. A circuit of depth 0 consists of only sources, and the circuit function is the identity map. A circuit of depth k consists of at least one device at depth k . For every such device \mathcal{D} , the circuit function assigns to \mathcal{D}_y the valuation obtained by applying the device function of \mathcal{D} to valuations of the outputs of the devices preceding \mathcal{D} . The circuit function of a simple circuit with a device \mathcal{D} is the device function of \mathcal{D} .

The dynamic interpretation of a circuit is given by the step-by-step evolution of the state machine, specifically the Markov chain, corresponding to the circuit. At every time step, the state of the circuit evolves as follows: first, all the sources are read to obtain their new outputs. Then, iteratively, for

any N -bit device \mathcal{D} at depth i the variables (including the output) are (probabilistically) updated using the outputs of all the devices and sources at depth less than i , by applying $Enq(b_1, \dots, b_N, d_1, \dots, d_N) \circ Deq \circ Time$ to the state of \mathcal{D} . Note that the values of the inputs b_1, \dots, b_N are fixed for \mathcal{D} as they are outputs from sources and devices at a lower depth (sub-circuit). The values of d_1, \dots, d_N are probabilistically chosen from the delay distributions of γ .

Example 3 Consider the device \mathcal{D}_1 of Example 1 connected to a 1-bit random source s_1 . We call this simple feed-forward circuit C_1 ; its circuit function $f_{C_1}(s_1) \triangleq f(s_1)$ equals $\neg s_1$.

Consider the circuit C_2 obtained by interconnecting devices $\mathcal{D}_2, \mathcal{D}_3$, with random sources s_1, \dots, s_4 at b_1, \dots, b_4 , respectively. This is a feed-forward circuit of depth 2 and circuit function $f_{C_2}(s_1, s_2, s_3, s_4) = f_2(f_1(s_1, s_2), s_3, s_4)$. ■

As mentioned above, the static and dynamic interpretations of a circuit are related when the sources are constant and time goes to infinity. More precisely, consider a circuit C with constant sources s_1, \dots, s_M fixed at values c_1, \dots, c_M and let π be the steady-state distribution of the Markov chain corresponding to the circuit C with these inputs. Then, for any device \mathcal{D} in C , at any state in the support π , the valuation of \mathcal{D}_y is the same as the valuation of \mathcal{D}_y in $f_C(c_1, \dots, c_M)$. This is proved in Theorem 3.

Of course, constant inputs and observing outputs as time goes to infinity, is not useful for performing computations quickly. Thus, in the next section we introduce a quantitative notion of correctness that allows us to observe the outputs of a stochastic circuit arbitrarily quickly, with some probability of error.

III. STOCHASTIC CORRECTNESS AND ITS ANALYSIS

Having introduced the notion of stochastic devices, circuits, and sources, we now proceed to define a meaningful quantitative notion of correctness for feed-forward stochastic circuits and show how this quantitative property can be computed and verified.

In a stochastic circuit, the computed output depends in the inputs and the stochastic delays. Thus, there is no *guarantee* that a correct output is observed for a given input. We measure the correctness of a circuit by the fraction of correct outputs observed in the long run. This is defined as the *correctness factor (CF)* of a circuit. CF of a stochastic circuit is defined with respect to an *observation period* $K \in \mathbb{N}$ which is the periodicity with which the output(s) are observed. This period K corresponds to the frequency with which signals are latched in the physical circuit; it is usually determined by a hardware clock (oscillator) that drives the timing-dependent parts of the circuit.

Definition 4: Given a circuit C with M 1-bit input sources s_1, \dots, s_M , $K \in \mathbb{N}$, and a designated output device \mathcal{D} in C , for each $q \in \mathbb{N}_{>0}$, C is said to be *correct with observation period* K if $\mathcal{D}_y(qK) = f_C(s_1((q-1)K), \dots, s_M((q-1)K))$, where f_C is the circuit function for \mathcal{D} . The total number

of times C is correct upto time t is denoted by $z_C(t)$. The correctness factor (CF_K) of C is:

$$\lim_{t \rightarrow \infty} z_C(t) / \lfloor \frac{t}{K} \rfloor.$$

Thus, at every observation time $t = qK$ the circuit is correct if the output matches with the output of the circuit function applied to the inputs from time $t - K = (q - 1)K$, and the correctness fraction is the fraction of correct observations as time goes to infinity. In this paper, whenever we consider correctness with observation period K , we implicitly assume that all the sources are K -period sources. Then, the K -period correctness factor CF_K of a simple circuit \mathcal{C} with a M -bit device \mathcal{D} and M 1-bit sources s_1, \dots, s_M depends on (a) the delay distributions $\{\Gamma_i\}_{i=1}^M$ of \mathcal{D} , (b) the distributions of the sources specified by the parameters $\{p_i\}_{i=1}^M$, and (c) the device function f of \mathcal{D} . We believe that an estimate of CF_K will guide the designer in choosing the trade-offs in clock (latching) frequency and the overhead due to the error in computations. This is directly related to other elements like voltage and power. Thus, CF_K will be one of the key entities required during the design of a stochastic circuit.

A. Properties of Stochastic Circuits

In this section, we prove several properties of stochastic circuits and ultimately derive expressions for correctness factors.

Calculating the correctness factor (CF_K) of an arbitrary circuit with arbitrary input sources involves finding the invariant distribution of the resulting Markov chain. We first analyze the correctness factor for simple circuits and then extend the analysis to general circuits. We begin with several basic properties of stochastic devices and circuits. Invariant 1 bounds the length of the internal queues in all devices.

Invariant 1: For any N -bit device \mathcal{D} in any circuit C , in any reachable state, for each $i \in [N]$, $queue_i.length \leq \max(\gamma_i)$.

Proof: Recall from Definition 1 that for *Enq* transition of the device \mathcal{D} , if an element $\langle b, d \rangle$ is inserted into $queue_i$, all $\langle b', d' \rangle$ where $d' > d$ will be removed from queue. Thus for each $d \in \mathbb{N}$, there can be at most one $\langle b, d \rangle$ in $queue_i$. Since $d \leq \max(\gamma_i)$, $queue_i$ can have at most $\max(\gamma_i)$ elements. ■ The next lemma states that a device connected to constant sources ultimately stabilizes to a fixed output. This lemma is used to prove Theorem 3 which states that the output of stochastic circuits with constant sources ultimately stabilizes to the output corresponding to the circuit function applied to the constant inputs.

Lemma 2: For a stochastic device \mathcal{D} with constant sources c_1, \dots, c_M , $\exists N \in \mathbb{N}$, such that $\forall t > N, \mathcal{D}_y(t) = f(c_1, \dots, c_M)$.

Proof: We have proved that $\forall i, queue_i.length \leq \max(\gamma_i)$ and at least one element is removed from $queue_i$ for each *Deq* transition. Also, since the inputs are constant sources we have that for each *Enq* transition, the tuple $\langle b, d \rangle$ enqueued has same value of b . Thus, after $\max(\gamma_i) + 1$ transitions of the system, for all $\langle b, d \rangle \in queue_i$, we have $b = c_i$. Thus, $\forall t > \max(\gamma_i) + 1, h_i(t) = c_i$.

For the device \mathcal{D} , $\forall t > \max\{\max(\gamma_1), \dots, \max(\gamma_M)\} + 1$, we have $\forall i, h_i(t) = c_i$. Thus $\mathcal{D}_y(t) = f(h_1(t), \dots, h_M(t)) = f(c_1, \dots, c_M)$. ■

Theorem 3: For a stochastic circuit C with constant sources c_1, \dots, c_M , $\exists N \in \mathbb{N}$, such that $\forall t > N, \mathcal{D}_y(t) = f_C(c_1, \dots, c_M)$, where \mathcal{D} is the designated output device.

Proof: We will prove this by induction on the depth of the circuit C . The base case is a circuit of depth 1 and it trivially follows from Lemma 2.

Inductive hypothesis: For all stochastic circuits of depth l , $\exists N \in \mathbb{N}$, such that $\forall t, t' > N, \mathcal{D}_y(t) = f_C(c_1, \dots, c_M)$, where \mathcal{D} is the designated output device.

Consider a stochastic circuit C of depth $l + 1$. It is built by combining several stochastic circuits of depth l , say C'_1, \dots, C'_r and a stochastic device \mathcal{D} , where the inputs of \mathcal{D} are the output of C'_1, \dots, C'_r or c_1, \dots, c_M . Let, N'_1, \dots, N'_r be the values after which the output from C'_1, \dots, C'_r is constant. Thus, $\forall t > \max\{N'_1, \dots, N'_r\} + 1$, the inputs to the device \mathcal{D} are constant sources. Thus, $\forall t, t' > \max\{N'_1, \dots, N'_r\} + 1 + \max\{\max_{\gamma_1}, \dots, \max_{\gamma_q}\} + 1, \mathcal{D}_y(t) = f_C(c_1, \dots, c_M)$. ■

Lemma 4: A stochastic device \mathcal{D} with Markovian input sources s_1, \dots, s_M is a finite state Markov chain.

Proof: A stochastic device \mathcal{D} is a collection of queues namely $queue_1, \dots, queue_M$. In Invariant 1 we have proved that $queue_i.length$ is finite and thus the state space of $queue_i$ is finite. Let $queue_i(t)$ be the state of $queue_i$ at time t . In order to prove this lemma, it is enough if we show that the $queue_i$ satisfies the Markovian property that $P(queue_i(t + 1) | queue_i(t), queue_i(t - 1), \dots) = P(queue_i(t + 1) | queue_i(t))$. We prove that Markovian property is satisfied on all possible transitions enabled from $queue_i$ and thus $queue_i$ satisfies the same.

- *Time:* This transition decreases the *deadline* component of each element in the queue deterministically.
- *Deq:* This transition removes the element in the queue with *deadline* = 0 deterministically.
- *Enq:* This transition inserts an element $\langle b, d \rangle$ in the queue by removing all the elements with *deadline* $\geq d$. Here, d is chosen from distribution γ_i and b is generated from a Markovian input source s_i . Thus, it preserves the Markovian property.

Hence, $queue_i$ satisfies the Markovian property and hence, a stochastic device \mathcal{D} with Markovian input sources s_1, \dots, s_M is a finite state Markov chain. ■

Remark: The Markov chain is time-homogeneous if all the sources in the circuit are time invariant (for example, random or constant), otherwise the chain is time-nonhomogeneous. It can also be shown that \mathcal{D} is irreducible and aperiodic, and therefore it has a unique stationary distribution².

B. Analysis of Steady State Correctness

Consider a simple circuit $C = (S, \mathcal{D})$, where $S = \{s_1, \dots, s_M\}$ is a set of Markovian input sources. We have

²The proof of this will be given in a future paper.

established that C is a finite state Markov chain. Correctness factors of such circuits can be derived by analyzing this Markov chain through probabilistic model checking or Monte Carlo simulations. In order to gain insight about the dependence of CF on various factors, in the remainder of this section, we present new methods for analytically calculating CF for simple circuits. Recall from Definition 4 that $CF_K = \lim_{t \rightarrow \infty} z_C(t) / \lfloor \frac{t}{K} \rfloor$. Thus $z_C(t)$ can be viewed as a counting process with the time scale $0, K, 2K, \dots$:

$$z_C(lK) = \begin{cases} z_C((l-1)K) + 1 & \text{if } \mathcal{D}_y(lK) = f(s_1((l-1)K), \\ & \dots, s_M((l-1)K)) \\ z_C((l-1)K) & \text{otherwise.} \end{cases}$$

In what follows, we show that the probability of $\mathcal{D}_y(lK) = f(s_1((l-1)K), \dots, s_M((l-1)K))$ can be calculated from γ_i , f and $\{p_1, \dots, p_M\}$ where $\{p_1, \dots, p_M\}$ are the parameters of $\{s_1, \dots, s_M\}$.

C. Calculation of Steady State Correctness

In this section, we arrive at a general expression for the correctness of an M -input simple stochastic circuit $C = (S, \mathcal{D})$. Informally, an observed output of such a circuit can be correct in three ways: (a) All the current inputs presented actually appear at the head of the queue before the outputs are latched or observed, (b) some of the current inputs do not appear at the head, but the corresponding previous input turns out to have the same value, and (c) some of the current inputs do not appear nor are the corresponding previous inputs the same, yet the output from the gate with these different input values turns out to be the same as the output corresponding to the current inputs. Theorem 6 derives the correctness factor of a circuit by analyzing the probability of each of these events.

We define $\text{delay}(\gamma, t)$ as the *deadline* value generated in *queue* according to the distribution γ at time t . Given $T, t \in \mathbb{N}$ and $t < T$, let $P(\gamma, T, t)$ denote the probability that $\forall t_1 \in (t, T], \text{delay}(\gamma, t_1) + t_1 > T$ and $\text{delay}(\gamma, t) + t \leq T$. That is, $P(\gamma, T, t)$ denotes the probability that the input appearing at time t experiences a delay such that it has the smallest deadline at time T and all subsequent inputs experience delays that make them come after time T .

Lemma 5: If $T > \max(\gamma)$, then $\sum_{t=1}^{T-1} P(\gamma, T, t) = 1$.

Proof: We define the set $S = \{t \in \mathbb{N} \mid \text{delay}(\gamma, t) + t < T\}$. Since $T > \max(\gamma)$, $0 \in S$, and so, $S \neq \emptyset$. Also, as $\min(\text{supp}(\gamma)) > 0$, for every $t_1 \in S$, $t_1 < T$. Now, let $t_{max} = \max(S)$. Since, $t_{max} \in S$, we have $t_{max} < T$. Further, $\forall t_2 \in (t_{max}, T], t_2 + \text{delay}(\gamma, t_2) + t_2 > T$. Thus, for $i < T$, $P[t_{max} = i] = P(\gamma, T, i)$, and $t_{max} < T$, thus $\sum_{i < T} P[t_{max} = i] = 1$ which implies that $\sum_{i=1}^{T-1} P(\gamma, T, i) = 1$ ■

By substituting $T = lK$ in Lemma 5, it follows that

$$\sum_{t < (l-1)K} P(\gamma, lK, t) = 1 - \sum_{t=(l-1)K}^{lK-1} P(\gamma, lK, t). \quad (1)$$

It is worth noticing that $\sum_{t=(l-1)K}^{lK-1} P(\gamma, lK, t)$ depends only on γ and K but not on the value of l (as $l \rightarrow \infty$). Next, we define a key quantity $\mathcal{E}(\gamma_i, K)$, which is the probability that at least one of the inputs over K time steps is latched or observed as the output. Thus, if we have sources with K clock period, the probability that $s_i((l-1)K)$ will be latched at $h_i(lK)$ is given by $\mathcal{E}(\gamma_i, K)$.

$$\mathcal{E}(\gamma, K) \triangleq \sum_{t=(l-1)K}^{lK-1} P(\gamma, lK, t) \quad (2)$$

This represents that there exists a time-step t , $(l-1)K \leq t < lK$, such that $\text{delay}(t)_i < lK - t$. Therefore, $1 - \mathcal{E}(\gamma_i, K)$ represents that $\forall t, (l-1)K \leq t < lK, \text{delay}(t)_i > lK - t$ and it follows that $1 - \mathcal{E}(\gamma_i, K) = \prod_{i=1}^K \Gamma(> i)$. Thus, $\mathcal{E}(\gamma_i, K) = 1 - \prod_{i=1}^K \Gamma(> i)$. Also, by definition, we have $\mathcal{E}(\gamma, 0) = 0$.

For a stochastic circuit, it is not always the case that the input $s_i((l-1)K)$ will be latched at $h_i(lK)$. However, in some cases it is possible that the output will be correct even when all the inputs are not latched properly. This probability of being randomly correct even when some of the current inputs do not appear at the head of the queue, is characterized by the quantity *random correctness probability (RCP)*.

Informally, *RCP* corresponds to the probability that two outputs from the same device with a set of common inputs and two sets of independent but identically distributed inputs, are the same. In other words, we fix a set of Markovian sources U and we have two sets of independent but identical Markovian sources S and S' . *RCP* gives the probability of the event that the output of the device \mathcal{D} with input $U \cup S$ is equal to the output of device \mathcal{D} with $U \cup S'$ as input.

Definition 5: Given a function $f : \mathbb{B}^M \rightarrow \mathbb{B}$, with common random sources $U = \{u_1, \dots, u_r\}$ with parameters p_1, \dots, p_r and two sets of independent sources s_{r+1}, \dots, s_M and s'_{r+1}, \dots, s'_M with the same parameters p_{r+1}, \dots, p_M , we define *random correctness probability (RCP) of f with respect to p_1, \dots, p_M and $U \subseteq [M]$* as:

$$\begin{aligned} RCP(f, p_1, \dots, p_M, U) &= \sum_{x \in \mathbb{B}^r} \sum_{z \in \mathbb{B}^{M-r}} \sum_{z' \in \mathbb{B}^{M-r}} I_{f(x,z)=f(x,z')} \\ &\times \prod_{i=1}^r P[u_i = x_i] \times \prod_{i=1}^{M-r} P[s_i = z_i] \times \prod_{i=1}^{M-r} P[s'_i = z'_i], \end{aligned}$$

where $I_{f(x,z)=f(x,z')}$ is the identity function that returns 1 only when $f(x, z) = f(x, z')$ and 0 otherwise.

Observe that *RCP* depends only on the circuit function f_C , the input parameters p_1, \dots, p_M and U . The RCPs for seven different types of elementary 2-bit stochastic devices are shown in Figure 3.

Theorem 6: The *correctness factor* of a simple circuit $C = (S, \mathcal{D})$ is given by the following equation

$$\begin{aligned} CF_K &= \sum_{Q \subseteq [M]} \left(\prod_{i \in Q} \mathcal{E}(\gamma_i, K) \times \prod_{j \notin Q} (1 - \mathcal{E}(\gamma_j, K)) \right) \\ &\times RCP(f, p_1, \dots, p_M, Q). \end{aligned}$$

	2 sources (.5, .5)			2 sources (.9, .1)		
	U = {a}	U = {b}	U = {a,b}	U = {a}	U = {b}	U = {a,b}
Const 0	1	1	1	1	1	1
AND	¼	¼	5/8	.838	.982	.8362
OR	¾	¾	5/8	.982	.838	.8362
XOR	½	½	½	.82	.82	.7048
NAND	¾	¾	5/8	.838	.982	.8362
a	1	½	½	1	.82	.82
NOT a	1	½	½	1	.82	.82

Fig. 3. Random Correctness Probabilities (RCP). The first column gives the names of the 2-bit circuits with inputs a and b . Columns 2-4 gives the RCPs of the circuits fed with random sources with parameters 0.5 and 0.5, with $U = \{a\}$, $U = \{b\}$, and $U = \{a, b\}$, respectively. Columns 5-7 gives the RCPs for the same circuits with random sources with parameters 0.9 and 0.1.

Proof: We start by calculating the probability with which $z_C(lK)$ increments its value. Recall from Definition 1 that $D_y(lK) = f(h_1(lK), \dots, h_M(lK))$, and hence $z_C(lK)$ will increment its value by 1 only if $f(h_1(lK), \dots, h_M(lK)) = f(s_1((l-1)K), \dots, s_M((l-1)K))$. Since the delays generated by γ_i are not always less than K , it need not be the case that $h_i(lK) = s_i((l-1)K)$. Let $Q \subseteq [M]$ such that $\forall i \in Q, h_i(lK) = s_i((l-1)K)$ and $\forall j \notin Q, h_j(lK) \neq s_j((l-1)K)$. Then, for $j \notin Q$, $h_j(lK)$ and $s_j((l-1)K)$ are drawn from the same source but are independent of one another. Hence, the probability that $f(h_1(lK), \dots, h_M(lK)) = f(s_1((l-1)K), \dots, s_M((l-1)K))$ is equal to $RCP(f, p_1, \dots, p_M, Q)$.

Now, in order to obtain the probability with which $z_C(lK)$ will increment its value, we sum the probabilities for all possible Q 's. That is, $\sum_{Q \subseteq [M]} (\prod_{i \in Q} \mathcal{E}(\gamma_i, K) \times \prod_{j \notin Q} (1 - \mathcal{E}(\gamma_j, K)) \times RCP(f, p_1, \dots, p_M, Q))$. We observe that this value is independent of l and thus the value of CF_K from Definition 4 is equal to this probability computed above. ■

Example 4 . We use Theorem 6 to derive the correctness factor for a simple circuit consisting of an 2-bit *AND* device and two 1-bit 1-period random sources s_1 and s_2 . Suppose $\gamma_1 = \gamma_2 = \gamma$, where γ is given as $\gamma(1) = \gamma(2) = \gamma(3) = \gamma(4) = 1/4$. Since $\gamma_1 = \gamma_2 = \gamma$, we have $\forall K, \mathcal{E}(\gamma_1, K) = \mathcal{E}(\gamma_2, K) = \mathcal{E}(\gamma, K)$. Also, suppose the source parameters $p_1 = p_2 = p$. In what follows we derive the correctness factor for an observation period of 1:

$$\begin{aligned}
CF_1 &= \mathcal{E}(\gamma, 1) \times \mathcal{E}(\gamma, 1) \times RCP(f, p, p, \{1, 2\}) \\
&+ (1 - \mathcal{E}(\gamma, 1)) \times \mathcal{E}(\gamma, 1) \times RCP(f, p, p, \{2\}) \\
&+ \mathcal{E}(\gamma, 1) \times (1 - \mathcal{E}(\gamma, 1)) \times RCP(f, p, p, \{1\}) \\
&+ (1 - \mathcal{E}(\gamma, 1)) \times (1 - \mathcal{E}(\gamma, 1)) \times RCP(f, p, p, \emptyset),
\end{aligned}$$

where $f(s_1, s_2) = s_1 \wedge s_2$. Calculating the value of $\mathcal{E}(\gamma, 1)$

and substituting it in the above equation, we get

$$\begin{aligned}
CF_1 &= (1 - \Gamma(> 1)) \times (1 - \Gamma(> 1)) \\
&\quad \times RCP(f, p, p, \{1, 2\}) \\
&+ (1 - \Gamma(> 1)) \times \Gamma(> 1) \times RCP(f, p, p, \{2\}) \\
&+ \Gamma(> 1) \times (1 - \Gamma(> 1)) \times RCP(f, p, p, \{1\}) \\
&+ \Gamma(> 1) \times \Gamma(> 1) \times RCP(f, p, p, \emptyset).
\end{aligned}$$

From Definition 5 it follows that $RCP(f, p, p, \{1, 2\}) = 1$. $RCP(f, p, p, \{2\}) = RCP(f, p, p, \{1\}) = 1 - 2(1-p)p^2$; $RCP(f, p, p, \{2\})$ represents the probability of the output being correct when the first bit in the AND gate is latched properly, whereas the second bit is generated from an independent identical source. $RCP(f, p, p, \{1\})$ represents the symmetric case. Through similar reasoning, $RCP(AND, p, p, \emptyset) = (1 - p^2)^2 + p^4$. Also, from the distribution γ , we have $\Gamma(> 1) = 3/4$. By substituting these values in the above expression, we get

$$\begin{aligned}
CF_1 &= 1/16 + 6/16 \times (1 - 2(1-p)p^2) \\
&+ 9/16 \times ((1 - p^2)^2 + p^4)
\end{aligned}$$

■

IV. CORRECTNESS OF ELEMENTARY CIRCUITS

In this section we analyze elementary feed-forward circuits that are composed of more than one device. The elementary circuits we consider have the following structure. The circuit consists of $M + 1$ devices. M of these devices $\mathcal{D}_1, \dots, \mathcal{D}_M$ are 1-bit devices with device functions f_1, \dots, f_M and delay distributions $\gamma_1, \dots, \gamma_M$. The $M + 1^{th}$ device \mathcal{D}_{M+1} has M inputs with device function f_{M+1} and delay distributions $\gamma'_1, \dots, \gamma'_M$. The output of each of $\mathcal{D}_1, \dots, \mathcal{D}_M$ is mapped to a i^{th} input of \mathcal{D}_{M+1} . Each of the inputs for $\mathcal{D}_1, \dots, \mathcal{D}_M$ are fed from s_1, \dots, s_M Markovian sources. In the remainder of this section, we refer to this circuit as $C = (S, \mathcal{D}_1, \dots, \mathcal{D}_M)$, where $S = \{s_1, \dots, s_M\}$.

For $M = 1$, C represents a circuit with two devices connected in a sequence and \mathcal{D}_1 connected to a Markovian source. It may appear that the sequential composition of \mathcal{D}_1 and \mathcal{D}_2 is equivalent (bisimilar) to a new device with delay distribution $\gamma_1 + \gamma'_1$ and device function $f_1 \circ f_2$, however, it is easy to check this is not the case. This type of simple composition rule breaks down because of the overwriting property of the devices.

In the following analysis, we will use the quantity defined below:

Definition 6: Given two delay distributions γ_1 and γ_2 , and $K \in \mathbb{N}$,

$$V(\gamma_1, \gamma_2, K) = \sum_{i=1}^K (\mathcal{E}(\gamma_1, i) - \mathcal{E}(\gamma_1, i-1)) \times \mathcal{E}(\gamma_2, K-i).$$

Informally, $V(\gamma_1, \gamma_2, K)$ represents the probability that the delay generated by γ_1 and the delay generated by γ_2 under composition is less than K and thus, the input generated at $(l-1)K$ will be latched as output at lK .

Next, we present a set of circuits and derive the expressions for *correctness factor*.

Theorem 7: The correctness factor of C with observation period K is:

$$CF_K = \sum_{Q \subseteq [M]} \left(\prod_{i \in Q} V(\gamma_i, \gamma'_i, K) \right) \times \prod_{j \notin Q} (1 - V(\gamma_j, \gamma'_j, K)) \times RCP(f_C, p_1, \dots, p_M, Q)$$

Proof: The proof given is similar to proof of Theorem 6. We begin the proof by calculating the probability with which $z_C(lK)$ will be incremented by 1. Recall from Definition 1 that $D_{M+1,y}(lK) = f_{M+1}(h'_1(lK), \dots, h'_M(lK))$. Thus, from Definition 4 $z_C(lK)$ will increment its value if $f_{M+1}(h'_1(lK), \dots, h'_M(lK)) = f_C(s_1((l-1)K), \dots, s_M((l-1)K))$. Since the delays generated by γ_i and γ'_i are not always less than K , it need not be the case that for some i , $h'_i(lK) = f_i(s_i((l-1)K))$. Let Q be the set such that $\forall i \in Q, h'_i(lK) = f_i(s_i((l-1)K))$. This can only happen if there exists $t \in [(l-1)K, lK)$, $delay(\gamma_i, t) < lK - t$ and there exists $t_1 \in [(l-1)K + delay(\gamma_i, t), lK)$, $delay(\gamma'_i, t_1) < K - t_1$. Recall, from Definition 2 that $\mathcal{E}(\gamma_i, K)$ represents that $\exists t \in [(l-1)K, lK)$, $delay(\gamma_i, t) < lK$. Thus, the probability that $h'_i(lK) = f_i(s_i((l-1)K))$ is nothing but $\sum_{j=1}^K (\mathcal{E}(\gamma_i, j) - \mathcal{E}(\gamma_i, j-1)) \times \mathcal{E}(\gamma'_i, K-j)$, which is equal to $V(\gamma_i, \gamma'_i, K)$. Thus, the value of $h'_i(lK) = f_i(s_i((l-1)K))$ with probability $V(\gamma_i, \gamma'_i, K)$.

Now, $\forall j \notin Q, h'_j(lK) \neq f_j(s_j((l-1)K))$. thus, $h_j(lK) = f_j(s_j(t))$ for some $t < (l-1)K$. In this case, the values of $s_j(t)$ and $s_j(lK)$ are drawn from the same source but are independent of one another. The probability that $f_{M+1}(h'_1(lK), \dots, h'_M(lK)) = f_C(s_1((l-1)K), \dots, s_M((l-1)K))$ is given by $RCP(f_C, p_1, \dots, p_M, Q)$.

Thus, the probability with which $z_C(lK)$ increments by 1 is obtained by the sum over all possible Q which is $\sum_{Q \subseteq [M]} (\prod_{i \in Q} V(\gamma_i, \gamma'_i, K) \times \prod_{j \notin Q} (1 - V(\gamma_j, \gamma'_j, K)) \times RCP(f_C, p_1, \dots, p_M, Q))$. We observe that this term is independent of l and thus the value of *correctness factor* of the circuit C from Definition 4 is equal to the above given expression. ■

Example 5 . Having derived an expression for correctness of a family of elementary circuits, we calculate its value for a specific circuit: The circuit C has two devices \mathcal{A}_1 and \mathcal{A}_2 with the same device function and delay distributions. The device function $f_1(b) = f_2(b) = \neg b$; and $\gamma_1 = \gamma_2 = \gamma$, where γ is given by $\gamma(1) = \gamma(2) = \gamma(3) = \gamma(4) = 1/4$. The output of \mathcal{A}_1 is mapped to the input of \mathcal{A}_2 and the input to \mathcal{A}_1 is fed from an Markovian input source s with parameter p . The circuit function $f_C(b) = f_2(f_1(b)) = b$. The expression for correctness for this circuit as given by Theorem 7 is as follows:

$$CF_2 = V(\gamma, \gamma, 2) \times RCP(f_C, p, \{1\}) + (1 - V(\gamma, \gamma, 2)) \times RCP(f_C, p, \emptyset).$$

It is clear from Definition 5 that $RCP(f_C, p, \{1\}) = 1$. $RCP(f_C, p, \emptyset)$ represents the probability that the output of the circuit corresponds to the correct output when drawn from an identical random source. Hence, $RCP(f_C, p, \emptyset) = p^2 + (1-p)^2$. Also, from Definition 6, we have

$$V(\gamma, \gamma, 2) = (\mathcal{E}(\gamma, 1) - \mathcal{E}(\gamma, 0)) \times \mathcal{E}(\gamma, 1) + (\mathcal{E}(\gamma, 2) - \mathcal{E}(\gamma, 1)) \times \mathcal{E}(\gamma, 0).$$

Since the value of $\mathcal{E}(\gamma, 0)$ equals 0, we have $V(\gamma, \gamma, 2)$ equals $\mathcal{E}(\gamma, 1) \times \mathcal{E}(\gamma, 1)$. In Example 4, we derived that $\mathcal{E}(\gamma, 1) = 1/4$, substituting this value on CF_2 , we get

$$CF_2 = 1/16 + 15/16 \times (p^2 + (1-p)^2). \quad \blacksquare$$

V. ANALYTICAL RESULTS FOR SIMPLE CIRCUITS

Informally, the correctness factor of a circuit with respect to an observation period $K \in \mathbb{N}$ is the fraction of time that the circuit produces correct results (given inputs that change every K time steps), in the long run. We have shown in Lemma 4, that if the input sources are random then the behavior of a stochastic circuit can be modeled as a Markov chain, and consequently correctness factors can be computed from the invariant distribution of this Markov chain. In order to calculate the *correctness factor*, then, one can employ existing tools for probabilistic model checking such as PRISM [9] and MRMC [10]. In fact, although not reported here, we have derived the correctness factors of several stochastic devices using the PRISM model checker and with our own stochastic circuit simulator, and these results match with the analytical results discussed below. In what follows we describe the numerical values of the correctness factors for several simple circuits, obtained from the analysis of Section III-B.

We consider simple circuits consisting of (a) single 2-bit devices with two different kinds of delay distributions, namely uniform and exponentially-decaying, and (b) random sources with different parameters. In these settings, we obtain a set of graphs from the numerical results that show how the correctness factor changes with the clock frequency. These graphs are consistent with our earlier experimental results(?) from detailed architectural simulations and also with our informal understanding of how correctness factor scales with clock frequency.

In the rest of this section, each delay distribution γ has support $\text{supp}(\gamma)$ equal to $\{1, \dots, 10\}$. An uniform delay distribution has equal probabilities $\frac{1}{10}$, for each i , and an exponentially decaying distribution has mean 5.5 and exponentially decaying probabilities around this mean and is truncated at 10.

Figure 4 shows how the CFs for seven 2-bit devices (same as the ones in Figure 3), change with increasing period (K) when they are fed from two random sources, each with a parameter of 0.5. That is, each of these sources produces a 1 with probability 0.5, at every K^{th} time step. First, observe that the CF for a constant device (for example, Const 0 for which the output is always 0 independent of the inputs) is 1 for all

periods. For all other, devices CF is (strictly) monotonically increasing with K before it becomes 1. This holds for all our experiments, and matches the intuition that slower the clock frequency, more correct the circuit. Next observe that there are three distinct “bundles” of curves in the plot: AND, NAND, OR, a, NOT a, and XOR. This corresponds to the three distinct values of RCP in this setting and corroborates Theorem 6.

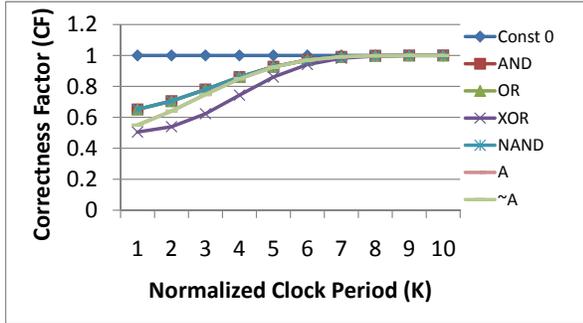


Fig. 4. CF_K variation with period K . Devices with uniform delay distributions and random sources with parameters 0.5 and 0.5.

Figure 5 shows similar set of results for random sources with 0.9 and 0.1. That is, the first source produces a 1 with probability 0.9 and the second source produces a 1 with probability 0.1, at every K^{th} time step. The graphs have similar characteristics as Figure 4, but we make the general observation that all other factors remaining the same the correctness factor is higher in this case. This matches with the informal notion that having biased sources of data can indeed help stochastic computation, compared to perfectly random sources.

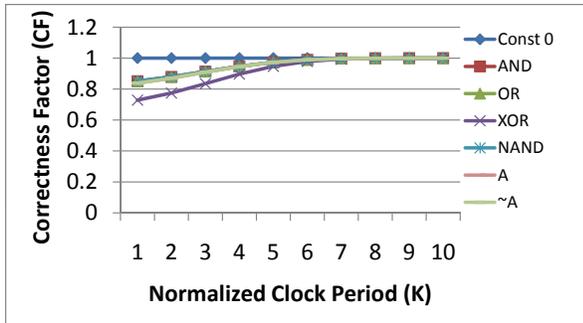


Fig. 5. CF_K variation with period K . Devices with uniform delay distributions and random sources with parameters 0.9 and 0.1.

Figure 6 shows the results for devices with exponentially-decaying delay distributions. Once again, the general observations made for Figure 4 hold for this case as well. Perhaps not surprisingly, these curves have similar extremal points as Figure 4, but they have a sharper “knee-point” in the middle. We anticipate that characterizing such knee-points will be central to designing stochastic circuits that are optimal for

certain types of sources and with respect to certain energy criteria. We plan on exploring this in the future.

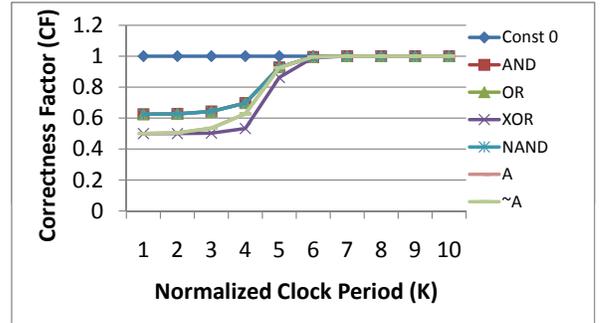


Fig. 6. CF_K variation with period K . Devices with exponentially-decaying delay distributions and random sources with parameters 0.5 and 0.5.

Finally, the results for devices with exponentially-decaying delay distributions and random biased sources are shown in Figure 7. These graphs share the characteristics of Figure 6 and Figure 5.

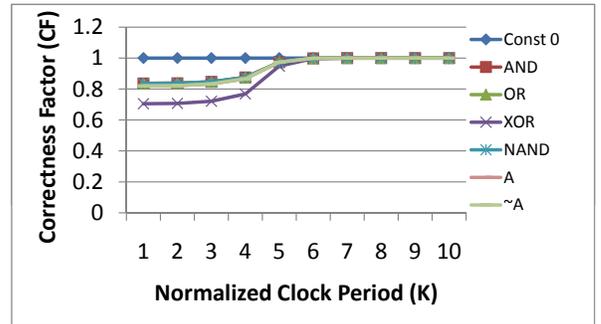


Fig. 7. CF_K variation with period K . Devices with exponentially-decaying delay distributions and random sources with parameters 0.9 and 0.1.

VI. RELATED WORK

The idea of computing with stochastically correct components is not new. Von Neumann studied the problem of reliable computing with unreliable devices [11]. Specifically, he characterized the system reliability of automata designed using stochastically correct three-input majority gates. The overhead of his constructions, however, are enormous. A number of later works also performed careful characterization of such constructions [12], [13], [14]. Other related theoretical work also includes the use of Markov random networks to design robust logic [15]. Such implementations have been shown to have large transistor counts, however. Finally, stochastic logic is proposed in [16] whereby Von Neumanns N-wire bundle presentation of Boolean variables is employed.

A large body of work exists on fault tolerance. N-modular redundancy (NMR) [17], for example, is a commonly employed fault-tolerance technique where computation is replicated in N processing elements, and the outputs are majority voted upon. The power and performance overhead of

NMR-based techniques are at least linear in N . Temporal redundancy-based techniques have been proposed as well. The performance overhead for such techniques can be significant. Techniques such as checkpointing [18], and coding techniques [19] have been proposed, each of which incur a significant energy-cost.

The work on stochastic processors [3], [4], [5] differs from the above in that the goal is not error avoidance at the processor-level. Rather the processor is allowed to produce errors. The errors are either tolerated by a hardware-based error resilience mechanism or they are propagated to the software stack where the software tolerates the errors. Since hardware and software-based error resilience techniques have overheads in terms of performance or output quality, a quantitative analysis needs to be done to balance the benefits of a stochastic processor design with the overheads of error resilience. The models and the results presented in this paper provide the framework and the tools necessary for making such design choices, albeit for simplistic circuits.

VII. CONCLUSIONS

We have presented a formal model for stochastic (feedback and feed-forward) circuits and developed a quantitative notion of correctness for the same. The building blocks of these circuits are stochastic devices, where the delays are generated randomly according to given distributions. We developed a methodology for constructing circuits by combining devices and sources of data, and presented formal semantics for stochastic feed-forward circuits. The quantitative stochastic correctness property, *correctness factor*, is a measure of fractional correct observations with respect to an observation period. We proved that the stochastic circuits are Markov chains (when fed from random sources) and derived closed form expressions for *correctness factor* of devices and a class of elementary circuits. Furthermore, we have presented numerical results obtained from the analysis, that illustrate the dependence of correctness on the delay distribution and the input distributions for several simple circuits.

This is the first step towards laying the foundations for quantitative reasoning about stochastic circuits and a lot of research remains to be done. Specifically, our results do not apply to circuits with feedback and analyzing such circuits will be the subject of a future paper. We believe that direct and exact analysis, like the one presented in this paper, for complex feedback-based circuits and processors is going to be challenging and hence one has to employ approximation and abstraction-based tools and techniques [20], [21], [22]. A different line of future research would be to evaluate different realizations of the same computational function with respect to energy consumption, clock-speed, and correctness factor. To this end, the results presented in this paper will provide the foundation for this direction of research.

REFERENCES

[1] "International technology roadmap for semiconductors 2008 update," 2008, available: <http://www.itrs.net/Links/2008ITRS/Home2008.htm>.

- [2] T. Austin, V. Bertacco, D. Blaauw, and T. Mudge, "Opportunities and challenges for better than worst-case design," in *Proc. ASPDAC 2005*, 2005, pp. 2–7.
- [3] R. Kumar, "Stochastic processors," in *NSF Workshop on Science of Power Management*, March 2009.
- [4] A. Kahng, S. Kang, R. Kumar, and J. Sartori, "Recovery-driven design: A methodology for power minimization for error tolerant processor modules," in *Proc. ACM/IEEE DAC 2010*, June 2010.
- [5] N. Shanbhag, R. Abdallah, R. Kumar, and D. Jones, "Stochastic computation," in *Proc. ACM/IEEE DAC 2010*, 2010.
- [6] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarazi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *In Proc. of ACM/IEEE DAC 2003*, 2003, pp. 338–342.
- [7] A. Kahng, S. Kang, R. Kumar, and J. Sartori, "Designing processors from the ground up to allow voltage/reliability tradeoffs," in *Proc. ACM/IEEE HPCA 2010*, January 2010.
- [8] —, "Slack redistribution for graceful degradation under voltage over-scaling," in *Proc. IEEE/SIGDA ASPDAC 2010*, January 2010.
- [9] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: A tool for automatic verification of probabilistic systems," in *Proc. 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, ser. LNCS, H. Hermanns and J. Palsberg, Eds., vol. 3920. Springer, 2006, pp. 441–444.
- [10] J.-P. Katoen, I. S. Zapreev, E. M. Hahn, H. Hermanns, and D. N. Jansen, "The ins and outs of the probabilistic model checker mrmc," in *QEST '09: Proceedings of the 2009 Sixth International Conference on the Quantitative Evaluation of Systems*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 167–176.
- [11] J. V. Neumann, "Probabilistic logics and the synthesis of reliable organisms from unreliable components," *Automata Studies*, pp. 43–98, 1956.
- [12] N. Pippenger, "Reliable computation by formulas in the presence of noise," *IEEE Trans. Info. Th.*, vol. 34, no. 2, pp. 194–197, 1988.
- [13] T. Feder, "Reliable computation by networks in the presence of noise," *IEEE Transaction Information Theory*, vol. 35, no. 3, pp. 569–572, 1989.
- [14] W. Evans and L. Schulman, "Signal propagation, with application to a lower bound on the depth of noisy formulas," in *Proc. of Annual Symp. on Foundations of Computer Science*, 1993, pp. 594–603.
- [15] K. Nepal, R. Bahar, J. Mundy, W. Patterson, and A. Zaslavsky, "Designing logic circuits for probabilistic computation in the presence of noise," in *in Design Automation Conf. (DAC)*, 2005, pp. 486–490.
- [16] W. Qian and M. Riedel, "The synthesis of robust polynomial arithmetic with stochastic logic," in *in Proc. of Design Automation Conf. (DAC)*, 2008, pp. 648–653.
- [17] N. Vaidya and D. Pradhan, "Fault-tolerant design strategies for high reliability and safety," *IEEE Trans. Comput.*, vol. 42, no. 10, pp. 1195–1206, 1983.
- [18] Y. Tamir, M. Tremblay, and D. Rennels, "The implementation and application of micro rollback in fault-tolerant vlsi systems," in *in Proc. of IEEE FTC*, 1988, pp. 234–239.
- [19] S. J. Piestrak, "Design of fast self-testing checkers for a class of berger codes," *IEEE Trans. Comput.*, vol. 36, no. 5, pp. 629–634, 1987.
- [20] J.-P. K. and Tim Kemna and Ivan Zapreev and David N. Jansen, "Bisimulation minimisation mostly speeds up probabilistic model checking," in *Tools and Algorithms for the Construction and Analysis of Systems, TACAS'07*, ser. LNCS, O. Grumberg and M. Huth, Eds., vol. 4424. Springer, 2007, pp. 87–101.
- [21] S. Mitra and N. Lynch, "Proving approximate implementation relations for probabilistic I/O automata," *Electronic Notes in Theoretical Computer Science*, vol. 174, no. 8, pp. 71–93, 2007.
- [22] F. van Breugel, M. Mislove, J. Ouaknine, and J. B. Worrell, "An intrinsic characterization of approximate probabilistic bisimilarity," in *Proceedings of FOSSACS 03*, ser. LNCS. Springer, 2003.