The Explicit Use of Probability in CMOS Designs and the ITRS Roadmap: From Ultra-low Energy Computing to a Probabilistic Era of Moore’s Law for CMOS*

Krishna V. Palem and Bilge E. S. Akgul
Georgia Institute of Technology
Atlanta, GA 30332
palem@ece.gatech.edu

1. Introduction

As CMOS device feature sizes scale down towards and into the “nano-regime,” their behavior is increasingly vulnerable to behaviors that are best characterized statistically. Such behaviors are induced both by the relative dominance of noise as the operating voltages scale down with feature sizes—see Kish [19] and Shepard [40] for example—as well as due to variations in device parameters that become increasingly hard to contain as outlined by Borkar et al. [4]. Historically, the semiconductor device as well as the microelectronics communities have invested in noise phenomenon with the intention of overcoming it. Thus, notable efforts in the context of coping with the former challenge—that of noise being a source of instability and hence resulting in behavior that is not deterministic in a digital design—are embodied in the interesting approaches of Hegde and Shanbhag [15], Ding and Mazumder [9], and Hernandez et al. [25], primarily aimed at yielding a degree of noise immunity (or noise tolerance) yielding “deterministic” digital circuits and hence computing platforms derived from them.

In this paper, we propose probabilistic CMOS (or pCMOS) as a novel approach to coping with the hurdles of technology scaling towards the “nano-regime.” Rather than treating noise as anathema to be overcome as a grossly unacceptable phenomenon, Palem [28, 30] outlined a framework for probabilistic switches and computational models, viewing noise as a “resource” whose controlled use might in fact be of value in digital designs, albeit probabilistic. These probabilistic computing frameworks were initially aimed at deriving low-energy computational platforms or architectures, for probabilistic algorithms as pioneered by Rabin, et. al [37]. This work and the associated energy characterizations were rooted within the framework of classical statistical thermodynamics (see Palem [31]). Thus, rather than viewing noise and its concomitant negative effects on deterministic computing as a grossly undesirable phenomenon, Palem’s work [29] articulated the need for characterizing the specific statistical properties of noise in detail. As a result, well-characterized noise was shown to be of value in realizing low-energy implementations of probabilistic algorithms, invariably associated with a well quantified probability of error; thus in this context, the low-energy instantiation of the algorithm could terminate with a well-quantified accuracy which is the probability with which it is correct.

* This work is supported in part by DARPA under Seedling # F30602-02-2-0124.
For example, in a probabilistic (or following Gelenbe [13], a “randomized”) neural network, the nodes (neurons) “fire” probabilistically in accordance with a Poisson distribution. By the virtue of a noise-induced probabilistic switch [8], these firing actions in a probabilistic neural network can be realized through truly probabilistic bits at a much lower energy cost, than the software-based pseudo random bits used if these networks are solved using a conventional microprocessor.

In order to extend these results and notions to the context of CMOS, the precise relationship between the probability of a switching step being correct $p$ and the associated physical attributes such as the operating voltage and the amount of noise in the system, were characterized by Cheemalavagu, Korkmaz, and Palem [7]. Subsequently, this work was extended significantly and systematized by Cheemalavagu, Korkmaz, Palem, Akgul, and Chakrapani [8]; the significance of this latter work, summarized in Section 2 of this paper, involves explicitly characterizing the relationship between noise and the associated probabilistic behavior that it induces in a CMOS based computing switch or device. Additionally, this work also characterized an explicit relationship between the probability $p$ with which the CMOS switch computes correctly, and its associated physical attributes such as the energy consumed by each switching step across two technology generations: these relationships were summarized succinctly in the form of two energy-probability relationships governing the behavior of probabilistic CMOS (or PCMOS) devices and switching.

In this report, we further consolidate and extend the above characterizations and laws by accounting for technology scaling in accordance with Moore’s law [26]. Specifically, the main theme of the work reported here is to characterize and evaluate the impact of technology scaling on the energy-probability relationships, characterized as “laws” governing the behavior of PCMOS “switches.” Following Palem [31], a switch is a computational primitive computing a one-input boolean function. Through this paper, the particular probabilistic switch of choice is a probabilistic inverter, ubiquitous to the design of digital circuits and systems.

We illustrate probabilistic behaviors induced by PCMOS at the device and switch levels in Figure 1, wherein the probability $p$ with which a device operates correctly is shown as the operating voltage $V_{dd}$ is varied. Here, a fixed amount of noise drives a CMOS inverter (switch) designed in a feature size of 0.25$\mu m$ using the TSMC process. In the sequel, we shall characterize relationship between $p$ and the switching energy; note that all of our characterizations and roadmap extensions will be in the context of the switching step of a probabilistic inverter (switch). They will be based on the source of noise being thermally induced on the inverter (switch) and the inverter’s interaction with noise being modeled using the classical foundations due to Stein [43]. For example, as seen from Figure 1, for a noise magnitude (RMS) value of 0.8V, the operating supply voltage $V_{dd}$—and hence the associated energy consumed—per a single switching step of PCMOS inverter increases exponentially with the probability $p$ of the switching step being correct. We will capture such characterizations as laws governing the behavior of probabilistic switching and we will present these laws in more detail in Section 2.

We claim that such a characterization is of value since current roadmaps including the influential ITRS roadmap for example, do not explicitly characterize the probability $p$ ($\frac{1}{2} < p < 1$) of the device’s correct execution and its associated physical attributes including its switching energy, as devices scale. To help characterize the benefits of using PCMOS devices, we introduce the novel notion of a noise oracle, in order to be able to quantify the limits to the (maximum) achievable energy and running-time gains when the source of probabilistic behavior is entirely free. We note in passing that this notion of an oracle is used in the same spirit to quantify “relative” (time) complexity of computations (see Yao [51] for example). We extend this notion here to encompass energy complexity of computations. If this is not the case and explicit energy
Figure 1. The relation between $p$ and $V_{dd}$ of a probabilistic inverter realized using TSMC 0.25μm technology with a noise magnitude (RMS) value of 0.8V.

and computational (time) investments are needed to realize probabilistic switching, then the gains so achieved will be lower than the amounts determined by these oracle based limits, which will be the topic discussed in Section 3.

Thus, our contribution described in the sequel can be viewed as adding a new dimension to the ITRS roadmap determined by the probability parameter $p$ at the level of a CMOS device and its associated switching step. A distinguishing feature of PCMOS technology is its ability to enable design of novel ultra low-energy architectures wherein $p$ serves as an explicit design parameter, on par with and influencing historically well-understood parameters such as throughput and switching energy. The impact of the device level behaviors on architectural realizations shall be the topic of Section 4.

On a related yet distinct note, probabilistic behaviors could also be induced at the increasingly small feature sizes due to variations in the device’s and hence the derived switch’s (say inverter’s) physical parameters across devices, as integrated circuits approach the Tera-scale (see Borkar et al. [4]). In this context, each switch behaves in a deterministic and sustained manner across successive switching steps when compared to its own past behavior—for example, its current output, operating voltage, threshold voltage and other such physical attributes stay invariant with time for any given switch—whereas these attributes can vary across individual switches as well as the interconnect in a Tera-scale IC (also see Burnett et al. [6], Nassif [27], and Venkatraman and Burleson [46]).

To contrast these notions of “probabilistic switching” with those associated with a probabilistic switch derived from PCMOS as described in the previous paragraphs and constitute the principal subject of this report (Sections 2, 3, and 4) wherein the probabilistic variations are temporally observable—the behavior of each individual switch affected by noise varies across successive switching steps probabilistically, we will refer to the type of probabilistic behaviors induced by parameter variations as being spatially observable. In Section 6, we will outline the relationship between and contrast temporally observable probabilistic switching realized through PCMOS devices with an emphasis on realizing ultra low-energy architectures for probabilistic applications, and “dual” notion of probabilistic behaviors that are spatially observed, caused by parameter variations. These relationships suggest methodologies for extending and
adapting the techniques developed within the context of temporally observed probabilistic switching to encompass those that are spatially induced, with the goal of deriving novel defect-tolerant computing architectures.

1.1. Roadmap of the Paper

The rest of this report is organized as follows. Section 2 presents the laws governing the behavior of a PCMS inverter and their validations through HSpice simulations across technology generations in the presence of a noise oracle, where the noise is freely available. Section 3 presents reconciliation of the behavior of a PCMS inverter with the absence of a noise oracle, where the cost of generating a noise source in the desired amount is also considered. Section 4 presents the extrapolation of device-level behaviors of PCMS to the architecture level realizations in the context of probabilistic applications, and Section 5 presents one other benefit of PCMS devices when quality of randomness also matters. Finally, Section 6 outlines space varying PCMS devices to cope with parametric variations.

2. The Laws Governing PCMS and the ITRS Roadmap

Recalling that our (inverter) switches realized using PCMS exhibit probabilistic behaviors in time induced by noise, our primary characterization (in Section 2.1) involves the interplay of two main attributes—the magnitude of noise quantified canonically as its RMS (root mean square) value and denoted by \( \sigma \), as well as the switching voltage denoted as \( V_{dd} \), both represented in the units of Volts—with the probability (of the inverter switch producing a “correct” output) \( p \). Based on this interplay, through Sections 2.1–2.3, we will characterize the relationship between the probability parameter \( p \) and the energy consumed during each switching step, in the form of two basic laws. Rather than depending on characterizations that are affected by two distinct attributes, namely the noise magnitude \( \sigma \) and the switching voltage \( V_{dd} \), by combining them, we show in Section 2.4 that the probability \( p \) with which a PCMS switch operates correctly across (Moore’s law) technology generations, is dependent on a single crucial attribute: its noise-to-signal ratio (NSR). Thus, the entire space of interest is captured by a unifying third law (Section 2.4) that simultaneously captures the relationship between the probability parameter \( p \), the noise magnitude \( \sigma \), and \( V_{dd} \) through the concept of NSR.

2.1. The ITRS Roadmap for PCMS Inverters with a (Probabilistic) Noise Oracle

A mathematical characterization of a probabilistic inverter (switch) can be expressed as follows:

\[
I(x) = \begin{cases} 
\pi & \text{with probability } p \\
x & \text{with probability } (1-p) 
\end{cases}
\]

Thus, with an input value of \( x = 0 \), the correct (and inverted) output value of 1 will occur with a probability \( p \), whereas the incorrect value 0 will occur on any particular invocation or switching step of the inverter with a probability of \( (1-p) \). The probability parameter \( p \) remains implicit in the above definition.

A mathematically equivalent definition of such an inverter is given as follows, wherein the probabilistic input \( \pi \) determines \( p \) as an explicit argument to the inverter function \( I \):

\[
I(x, \pi) = \begin{cases} 
\pi, & \pi = 0 \\
x, & \pi = 1 
\end{cases}
\]
Figure 2. A realization (a) of a probabilistic inverter using a thermal noise based oracle coupled to the input and (b) its mathematical behavior.

\[
\pi = \begin{cases} 
0 & \text{with probability } p \\
1 & \text{with probability } (1 - p) 
\end{cases}
\]

In the latter realization, the random variable \( \pi \) is an input to the “inverter” function \( \mathcal{I} \) and causes the inverter to function correctly whenever \( \pi = 0 \), which occurs with a probability of \( p \). The inverter output is incorrect whenever \( \pi = 1 \) with an associated probability value of \( (1 - p) \). A physical realization of a probabilistic inverter with a thermal noise source is shown in Figure 2. In this realization, the thermal noise with an “adequate” value of noise magnitude \( \sigma \) is provided through the use of a noise oracle, denoted as \( \mathcal{O} \) that serves as the basis for realizing the probabilistic variable \( \pi \). Here, a noise oracle for producing the random variable \( \pi \) with an associated probability parameter \( p \) is a function \( \mathcal{O} : \{0, 1\} \rightarrow \pi \in \{0, 1\} \) such that \( \pi = 0 \) with a probability \( p \). The oracle function \( \mathcal{O} \) can be consulted instantaneously and by definition, yields \( \pi \) with no (zero) expenditure of energy.

To reiterate, informally, the physical realization that is shown in Figure 2 is said to use a noise oracle for determining \( \pi \) whenever the probabilistic noise source is available for free at the requisite value of \( \sigma \), to yield a probabilistic inverter with the associated probability value of \( p \); in other words, there is no energy or time needed to produce \( \pi \) and furthermore, in an idealized setting, \( \pi \) is determined instantaneously. Thus, the noise oracle \( \mathcal{O} \) can be consulted for free and yields the random variable \( \pi \) instantaneously. By not paying for the production of the variable \( \pi \), we can determine the maximum amount of benefits or savings, both in the performance of a probabilistic application and in its energy consumption. An outline of these benefits both in the presence of a noise oracle which yields the variable \( \pi \) for free, and a comparison of this case to one where it costs energy and time to produce \( \pi \), will be the topic of Section 3. We will now start with an analytical characterization of an idealized PCMOS switch to serve as a backdrop for the empirical findings and behavioral characterizations in the subsequent sections.

2.1.1. An idealized PCMOS switch and the Oracle \( \mathcal{O} \)

Following Stein [43], a single switching step of a deterministic inverter is shown in Figure 3(a). In this figure, the transition from a value of 1 at the output of the inverter to a value of 0 is idealized and hence is instantaneous and occurs at a value of \( \frac{V_{dd}}{2} \). Moreover, the binary value of 0 is associated with a (measured) output voltage in the interval \( (-\infty, \frac{V_{dd}}{2}) \) and similarly the binary value of 1 is associated with the complementary interval \( [\frac{V_{dd}}{2}, +\infty) \). Also following Stein [43], noise can be modeled as being coupled with the output of the inverter as shown in
Figure 3. (a) An idealized inverter and (b) its probabilistic behavior determined by thermal noise coupled at its output.

Figure 3(b); thus, it affects the output value probabilistically as follows. As it interacts with the voltage values representing a deterministic 0 or 1 corresponding to the “signal,” noise which is characterized by a Gaussian distribution with a standard variation \( \sigma \) is superimposed on the signal. The mean of the noise distribution corresponds to the voltages 0 and \( V_{dd} \) as shown in Figure 3(b).

Stein models the noise as being additive to the signal, which serves as a basis for us to develop a probabilistic inverter derived by combining thermal noise with the output of a deterministic inverter. For example, with an output value of 1—measured to be \( V_{dd} \) in the ideal case—the instantaneous value will be entirely determined by the noise distribution. Thus, while the output value ought to be \( V_{dd} \), because of additive noise, it can easily be in the interval \( (-\infty, \frac{V_{dd}}{2}) \) inherently yielding a value of 0 and the associated error probability \( (1 - p) \) is exactly the area labeled A in Figure 3(b); a symmetric argument yields the probability of an output value of 0 being erroneously treated to be 1 and corresponds to the area labeled B (Figure 3(b)).

With this as background, the oracle \( \mathcal{O} \) can be defined as a function, informally characterized through Table 1 with two input arguments \( i \) and \( \eta \) and one output \( o \). Here, \( i \) denotes the input to the inverter and \( \eta \) denotes the (additive) noise magnitude at the output node of the inverter. For example, considering the first row of Table 1, when the input to the inverter is a (deterministic) value of 0, the output is correct and a value of 1 whenever the additive noise value \( \eta \geq -\frac{V_{dd}}{2} \). This follows from the simple observation that with an input value of 0, by definition, the idealized output value is \( V_{dd} \). Now, when \( \eta \geq -\frac{V_{dd}}{2} \), the observed or measured output value \( (V_{dd} + \eta) \) lies within the interval \( [\frac{V_{dd}}{2}, \infty) \) corresponding to an output of digital 1. Thus, in this case, \( o = 1 \). Similarly, with the same input and referring to the second row of Table 1, we note that when \( \eta < -\frac{V_{dd}}{2} \), the output value will be observed to be in the interval \( (-\infty, \frac{V_{dd}}{2}) \) leading to an incorrect output of digital 0—hence \( o = 0 \). The case where the input \( i = 1 \) is similarly developed in the remainder of the table, and can be easily understood following the symmetry with the case wherein \( i = 0 \).

2.1.2. The analytical model of an idealized probabilistic PMOS switch

With these definitions as background, we will now succinctly characterize the relationship between the probability \( p \), the operating voltage \( V_{dd} \) and the noise magnitude \( \sigma \) through the following basic analytical relationships, which we will refer to as an analytical model of a PMOS inverter given in Equations (1), (2) and (3) below. Note that in these equations, \( \text{inverf} \) refers to the inverse of the well-known error function [44].
<table>
<thead>
<tr>
<th>$i$</th>
<th>$\eta$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\geq -\frac{V_{dd}}{2}$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$&lt;-\frac{V_{dd}}{2}$</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>$&lt;\frac{V_{dd}}{2}$</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$\geq \frac{V_{dd}}{2}$</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1. Probabilistic inverter function characteristic with the inputs $i$ and noise magnitude $\eta$, and the output $\phi$.

![Graphs showing the relationship between $V_{dd}$ and $p$ with a noise RMS value $\sigma = 0.8$ Volts across four technology generations estimated analytically.](image)

**Figure 4.** The relationship between $V_{dd}$ and $p$ with a noise RMS value $\sigma = 0.8$ Volts across four technology generations estimated analytically.

**Analytical Model of a Probabilistic Inverter**

\[ p = 0.5 + 0.5 \cdot erf\left(\frac{V_{dd}}{2\sqrt{2} \cdot \sigma}\right) \]  

(1)

\[ \sigma = \frac{V_{dd}}{2\sqrt{2} \cdot \text{inverf}(2 \cdot p - 1)} \]  

(2)

\[ E = 4 \cdot C \cdot \sigma^2 \cdot [\text{inverf}(2 \cdot p - 1)]^2 \]  

(3)

The above analytical model allows us to deduce relationships between the probability parameter $p$, the voltage, or equivalently signal magnitude $V_{dd}$, the noise magnitude $\sigma$ and finally the energy consumed per switching step denoted by the symbol $E$. We will first state the two primary relationships stated as facts, that relate $V_{dd}$ and $\sigma$ respectively, to the probability parameter $p$. Subsequently, in Section 2.3, we will verify these analytical estimates through HSpice simulations of a PMOS inverter.

2.1.3. The two facts of a probabilistic inverter

Based on our analytical model described above, we now introduce the first fact relating $p$ to $V_{dd}$ and illustrate it across four technology generations in Figure 4.

**Fact 1** For a fixed value of $\sigma$, $p$ increases exponentially as $V_{dd}$ is increased.

The above fact follows immediately from the relationship in Equation (1) of the analytical model. Fact 1 can also be intuitively deduced from Figure 3(b): As $V_{dd}$ is decreased, while
Figure 5. The relationship between $\sigma$ and $p$ estimated analytically across four technology generations, with a $V_{dd}$ value that is the nominal operating voltage for the particular technology generation used.

keeping $\sigma$ fixed, as shown in Figure 3(b), the area A (or B) increases. This area corresponds to the probability of error $(1 - p)$. Thus, increasing the area A (or B) decreases $p$.

The second fact illustrated in Figure 5 across four technology generations follows from Equation (2).

Fact 2 For a fixed value of $V_{dd}$, $p$ decreases exponentially as $\sigma$ is increased.

Fact 2 can also be intuitively deduced from Figure 3(b). As $\sigma$ is increased while keeping $V_{dd}$ fixed, the intersection area A (or B) increases, resulting in a decrease of $p$.

2.2. The Two Energy-probability Laws

Continuing and based on the above two facts, we can further extend our characterizations of PMOS inverters through the following two central laws that guide their behavior in the context of the energy consumed, specifically as the probability parameter $p$ is varied. To reiterate, these laws enable us to directly characterize the relationship between the critical parameter of interest in the context of a switching step: the energy consumed, and the probability parameter $p$. We describe them below as the energy-probability laws. Both of these laws are direct and quantitative consequences of the relationship in Equation (3) of the analytical model.

The first law stated below relates the energy consumed in each switching step of a PMOS inverter to $p$ given a fixed amount of noise magnitude $\sigma$. Thus, as shown in Figure 6, the first law results in changes to the probability $p$ by varying the $V_{dd}$ value and hence results in varying values of $E$ determined by Equation 3 of the analytical model; recall that in this context, the amount of noise magnitude $\sigma$ is fixed ($\sigma = 0.8$V in Figure 6).

Law 1 For a fixed technology (feature size) generation, the energy consumed by a switching step of a probabilistic inverter increases exponentially as the probability of correct switching $p$ increases whenever the noise magnitude $\sigma$ remains a constant.

Akin to the first law, the second law stated below relates the energy consumed in each switching step of a PMOS inverter to $\sigma$, given a fixed value of $p$. Thus, as shown in Figure 7, the second law yields changes to the energy $E$ by varying the noise amount $\sigma$, for a fixed value
Figure 6. The first energy-probability law relating the switching energy to the value of $p$ across technology generations for a fixed value of noise magnitude, $\sigma = 0.8V$.

Figure 7. The second energy-probability law relating the switching energy to the value of $\sigma$ for fixed values of $p$ for 0.5$\mu$ AMI, 0.25$\mu$ TSMC, 65$nm$ and 90$nm$ IBM processes.

of probability $p$. In Figure 7, the four instances of the law are indicated, respectively, for 0.5$\mu$ AMI, 0.25$\mu$ TSMC, 65$nm$ and 90$nm$ IBM processes in a logarithmic scale for energy per switching step. Akin to the analysis in the context of the first law, it is easy to deduce the quadratic growth in energy as the noise magnitude $\sigma$ is increased.

**Law 2** For a fixed value of $p$ and for any particular technology generation, the energy consumed during a switching step of a probabilistic inverter (switch) increases quadratically with increasing $\sigma$.

This quadratical relationship between $\sigma$ and $E$ for a fixed value of $p$ as stated by Law 2 above, follows from the fact that the switching energy given as $E = \frac{1}{2} \cdot C \cdot V_{dd}^2$ is quadratically related to $V_{dd}$ which is linearly dependent on $\sigma$ (as shown in Equation 2).

We note in passing that the two primary relationships summarized as laws above clearly demonstrate that the probability parameter $p$ is affected by the value of the switching voltage
$V_{dd}$ as well as by the noise magnitude $\sigma$. We will now validate these analytical projections through validations using HSpice models.

**Remark:** From the analytical model (in Section 2.1.2), both the primary relationships (in Section 2.1.3) and the energy-probability laws (in Section 2.2) can be extrapolated over successive technology generations by choosing and substituting the corresponding load capacitance value $C$ from the ITRS roadmap [17] in the model.

### 2.3. Validating the Analytical Relationships and Laws

In this section, we will establish the validity of the crucial trends characterized as facts and as laws respectively in Sections 2.1.3 and 2.2. In Sections 2.3.1 and 2.3.2 below, we relate the analytical estimates of the two facts and of the two laws respectively, to their HSpice-simulated counterparts.

#### 2.3.1. Validating facts

The first fact shown in Figure 8 relates the probability parameter $p$ to the operating voltage $V_{dd}$ across technology generations. In this figure, the analytical and the simulated trends are shown for four technology generations, respectively 0.5μ AMI, 0.25μ TSMC, 90n and 65n IBM proprietary technologies. Thus, as validated in Figure 8, to realize different values of $p$ (given a fixed value of $\sigma$), Fact 1 (based on Equation 1) can be used to determine the amount of voltage ($V_{dd}$) scaling needed.

In order to evaluate the quality of the analytical model, beyond noting that its projections visually correspond well with the simulated values, we will also consider the minimum and maximum variations between the (analytically) estimated and the simulated values of $p$ in Table 2. As seen from the table as well as Figure 8, the analytical estimates closely follow the simulated values of $p$ as $V_{dd}$ is varied within a very small % deviation, and in all cases, the variations are less than 2.5%, for all four technology generations.

Moving to Fact 2, the corresponding reconciliation of the analytically anticipated values with their simulated counterparts is shown in Figure 9, again across all four technology generations of interest. The evaluation of the quality of the model in the context of Fact 2 is shown in Table 3 wherein the minimum and maximum variations between the (analytically) estimated and the simulated values for the corresponding characterizations of Fact 2 are tabularized. Recall that the deviations in the values of $p$ are realized by varying $\sigma$, for a $V_{dd}$ value fixed at the nominal operating voltage for each technology generation. In each case, maximum value of these variations are a remarkably low 0.52%.
### Table 2

<table>
<thead>
<tr>
<th></th>
<th>AMI 0.5µ</th>
<th>TSMC 0.25µ</th>
<th>IBM 0.09µ</th>
<th>IBM 0.065µ</th>
</tr>
</thead>
<tbody>
<tr>
<td>% max variation</td>
<td>2.2</td>
<td>2.48</td>
<td>0.67</td>
<td>0.5</td>
</tr>
<tr>
<td>% min variation</td>
<td>0.002</td>
<td>0.07</td>
<td>0.02</td>
<td>0.09</td>
</tr>
<tr>
<td>% avg variation</td>
<td>0.6</td>
<td>1</td>
<td>0.5</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Table 2. Quantifying the variations between the simulated and the analytically projected values of $p$ as $V_{dd}$ is varied, characterized as Fact 1.

### Figure 9

Fact 2 validated over 0.5µ AMI, 0.25µ TSMC, 90nm and 65nm IBM proprietary technologies.

#### 2.3.2. Validating the two probability-energy laws governing PCMOS switching

Turning our attention to the two crucial laws relating the energy consumed during a switching step and the associated probability parameter $p$, we will once again quantify the deviations between the analytical estimates from Section 2.2 and the corresponding behaviors determined through simulation studies. For completeness, we recall that in all of these comparisons, the probabilistic behavior is deemed to be free and is realized through a noise oracle (Section 2.1).

Following the style adopted in comparing the analytically predicted and the simulated values in the context of the two facts discussed above, we will first consider Law 1 in Figure 10. As shown in this figure, the analytically estimated values are matched very closely. Once again, the figure-of-merit that is of interest here is the maximum, minimum and the average variations between the two instances—namely, the analytically modeled and the HSpice-simulated characterizations—of the first law; we summarize these variations in Table 4. Here, the difference between the analytical model and the simulation results of the energy per switching step are higher than those presented in the context of Facts 1 and 2 and stem from the following reason. In the analytical model presented in Section 2.1.2 and used to estimate the energy consumption as $p$ is varied, we considered only the switching energy; this is, the energy consumed to charge/discharge the output load capacitance of the probabilistic inverter. However, in the

### Table 3

<table>
<thead>
<tr>
<th></th>
<th>AMI 0.5µ</th>
<th>TSMC 0.25µ</th>
<th>IBM 0.09µ</th>
<th>IBM 0.065µ</th>
</tr>
</thead>
<tbody>
<tr>
<td>% max variation</td>
<td>0.52</td>
<td>0.5</td>
<td>0.52</td>
<td>0.48</td>
</tr>
<tr>
<td>% min variation</td>
<td>0</td>
<td>0</td>
<td>0.02</td>
<td>0.03</td>
</tr>
<tr>
<td>% avg variation</td>
<td>0.15</td>
<td>0.17</td>
<td>0.2</td>
<td>0.24</td>
</tr>
</tbody>
</table>

Table 3. Quantifying the deviations between the simulated and the analytically projected variations of the value of $p$ as $\sigma$ is varied, characterized as Fact 2.
Figure 10. Law 1 validated over 0.5μ AMI, 0.25μ TSMC, 90n and 65n IBM proprietary technologies.

<table>
<thead>
<tr>
<th>% max deviation</th>
<th>AMI 0.5μ</th>
<th>TSMC 0.25μ</th>
<th>IBM 0.09μ</th>
<th>IBM 0.065μ</th>
</tr>
</thead>
<tbody>
<tr>
<td>% min deviation</td>
<td>4.4</td>
<td>3.9</td>
<td>25.4</td>
<td>35.8</td>
</tr>
<tr>
<td>% avg deviation</td>
<td>2.8</td>
<td>3.2</td>
<td>20</td>
<td>25.2</td>
</tr>
</tbody>
</table>

Table 4. Quantifying the deviations between the analytically projected and simulated variations of energy as they determine p (for a fixed σ) characterized by Law 1.

The context of the simulated estimates, we considered the total energy consumption, which includes the switching energy as well as the short circuit energy and energy consumed due to leakage. As can be seen from the table, the variations are higher for 90n and 65n processes, since the short circuit energy constitutes a larger portion of the total energy for the inverters designed in these two technologies. As is known widely, the short circuit energy is mainly caused by high rise and fall times of the input signal, the asymmetric rise and fall times, and a low output capacitance. For example, the higher the rise and fall times, the higher the switching duration and hence higher the short circuit energy dissipation observed during switching.

To factor in the effect of short circuit energy, we have enhanced the analytical model to include a short circuit energy model (the interested reader is referred to Korkmaz [20] for details). Using this short circuit model, the variations between the analytical and simulated values of E decrease as shown in Figure 11. For example, for an inverter (with an input signal having high rise and fall times) designed using TSMC 0.25μ process shown in Figure 11, the average deviation in E decreases from 49% (point A in Figure 11) down to 3% (point B in Figure 11), when the analytical model is extended to include short circuit energy.

One other reason for the % variations in E as seen from Table 4 to be larger for 90n and 65n processes when compared to 0.5μ and 0.25μ processes is due to the fact that the analytical model does not include the leakage energy consumption which constitutes a larger portion of the total energy in smaller feature sizes. Therefore, one aspect of our future work is to extend our analytical model to include the leakage energy consumption as well.

Considering Law 2, which was analytically characterized in Figure 7, and comparing these estimates with the values derived from HSpice simulations shown in Figure 12, we observe that the analytical model tracks simulated results well for all of the four technology generations. Akin to the trends discussed above in the context of Law 1, the deviations shown in Table 5 are larger for inverters designed in 65n and 90n processes, and once again, the difference is due to short circuit as well as the leakage energies. The maximum, minimum and average deviations between the analytical and the empirical results for the E values for the four technology generations can be seen from Table 5.
Figure 11. Comparing the simulated results with the analytical results when the analytical model (a) does not include the short circuit energy and (b) includes the short circuit energy model.

<table>
<thead>
<tr>
<th></th>
<th>AMI 0.5μm</th>
<th>TSMC 0.25μm</th>
<th>IBM 0.09μm</th>
<th>IBM 0.065μm</th>
</tr>
</thead>
<tbody>
<tr>
<td>% max deviation</td>
<td>4.4</td>
<td>3.9</td>
<td>25.4</td>
<td>33.5</td>
</tr>
<tr>
<td>% min deviation</td>
<td>0.5</td>
<td>2.4</td>
<td>16.8</td>
<td>20.2</td>
</tr>
<tr>
<td>% avg deviation</td>
<td>2.8</td>
<td>3</td>
<td>20.7</td>
<td>23.4</td>
</tr>
</tbody>
</table>

Table 5. Quantifying the deviations between the analytically projected and simulated variations of energy as they determine σ (for a fixed p) characterized by Law 2.

Figure 12. Law 2 validated over 0.5μ AMI, 0.25μ TSMC, 90n IBM proprietary technologies.
2.4. The Noise-to-signal Ratio (NSR) and the Unifying Third Law of PCMOS

The two laws relating probability and noise magnitude to energy were characterized in the previous section. The space of variations in the energy expended and the associated probability $p$ that these laws characterize were based on three independent parameters: the operating voltage $V_{dd}$ henceforth referred to as the signal, the noise magnitude $\sigma$ and the technology generations. It is important to note that the probability parameter $p$ and the energy $E$ whose relationship was characterized by these laws, are not independent parameters. In the context of the first law, the relationship between $p$ and $E$ was determined by varying the voltage $V_{dd}$. Thus, whereas the first law spanned two of these (three) dimensions by keeping the value of $\sigma$ fixed (see Figures 6 and 10), the second law also spanned two dimensions and characterized effect of changing values of $\sigma$ with $V_{dd}$ (implicitly), while keeping the $p$ values fixed (see Figures 7 and 12). The respective graphical characterizations of these laws illustrated in Figures 10 and 12 were depictions of these laws wherein the dimension being varied was represented as a parameter: for example, in Figure 10, keeping the value of $\sigma$ fixed, $V_{dd}$ was the parameter that was varied, which in turn uniquely defined a value of $p$ and $E$.

Let us now consider an alternate and more succinct form for the analytical model that describes $p$ as developed below. Rather than specifying the magnitudes of the signal and noise, respectively $V_{dd}$ and $\sigma$ as two independent variables, they are presented as a single ratio as a variation of the analytical model (see Equation 4 below).

Given that $V_{dd} = 2\sqrt{2 \cdot \sigma \cdot \text{inverf}(2 \cdot p - 1)}$, from Equation (2), we can express Noise-to-signal Ratio (NSR) as follows.

\[
NSR = \frac{\sigma}{V_{dd}} = \left[2\sqrt{2 \cdot \text{inverf}(2 \cdot p - 1)}\right]^{-1}
\]  
(4)

Thus, as shown in Figure 13, two of the three independent dimensions—the signal and noise magnitudes, respectively $V_{dd}$ and $\sigma$—are combined into a single dimension using the NSR through which we observe a surprising invariance in the behavior of $p$ across technology generations, certainly for an idealized probabilistic inverter characterized analytically above.

Law 3 In any technology generation, the probability parameter $p$ is uniquely characterized by the NSR value and decreases exponentially (as given in Equation 4) with the NSR value.

This crucial and unique relationship between the value of $p$ and that of NSR is shown in Figure 14 for all of the four technology generations based on HSpice simulations. It can be seen that the analytical formulations of this law illustrated in Figure 13(b) and the simulated variants match each other extremely well. We also note in passing that the analytical relationship of this law shown as Equation 4 is the same for all of the four technology generations since, the value of $p$ is independent of any technology dependent attributes such as the capacitance. This form of the relationship between the three independent ($\sigma$, $V_{dd}$, and technology generation), and the dependent parameter ($p$) of the model of a probabilistic inverter is very interesting for the following reasons. First, it allows us to establish a succinct relationship between all of the independent parameters and the probability parameter $p$ in a manner that does not depend on the CMOS technology generation represented typically as a feature size. As a result, and secondly, it provides a two-dimensional space that is easy to use and comprehend.
Figure 13. $V_{dd}$-noise RMS-probability relationship is reduced into a two-dimensional, technology independent NSR-probability relationship.

Figure 14. NSR-probability relationship validated across four technology generations.

To understand this, we contrast the “equivalent” but more complex representation shown in Figure 13(a) with Figures 13(b)—the latter simultaneously illustrating the invariance across technology generations and the concomitant simplicity of estimating $p$ given an NSR value. It is sufficient to remark that the maximum deviation between the simulated and the analytically estimated values occurs for a feature size of $0.5\mu$, and is no more than 8%.

3. Available Noise and the Noise Oracle

In all of the above characterizations, the requisite amount of noise RMS, $\sigma$, is deemed to be readily available and hence “free.” The fact that noise is available in increasing magnitude is justified by the fact that it is an ever increasing threat or an impediment to reliable switching as device feature sizes approach the nanometer scale. However, currently available magnitudes of $\sigma$ at coarser technologies (in device feature sizes as low as $65\text{nm}$ for example) are not adequate
Figure 15. In the absence of a noise oracle, (a) characterizing the amount by which the available noise magnitude and hence NSR are less than desired and (b) characterizing a resistor as a source of thermal noise.

to yield particular and specific values of \( p \) desired by the probabilistic application. Nevertheless, in order to validate the behaviors and benefits of PCMOS in technologies currently available—from 0.5\( \mu \)m down to 65nm being the scope of study in this report—we adopt the approach of increasing the noise magnitude value \( \sigma \) through amplification, such that the requisite noise-to-signal ratio and concomitant value of \( p \) is realized. The particular low-energy subthreshold amplifier’s structure and its attributes can be found in the work of Cheemalavagu, Korkmaz, Palem, Akgul, and Chakrapani [8]; its impact on the trends discussed in Section 2 will be the topic of this section.

All of the estimates and the characteristics in the previous section were contingent on the availability, at no extra (energy or time) cost, of a source of randomness of the “appropriate” magnitude given a probability parameter \( p \); note that Law 3 is a characterization of this relationship. However, noise at the requisite magnitude quantified by its RMS value, might not always be available. To illustrate this idea, we consider a thermal noise source derived from a resistor of 10 G\( \Omega \) as shown in Figure 15(a).

In Figure 15(b), we show the nominal \( V_{dd} \) value scaling down through technology generations, and the corresponding noise RMS (\( \sigma \)) needed to induce a probability parameter \( p = 0.85 \). This value of \( p \) is interesting since an inverter with \( p = 0.85 \) is used by a Bayesian inferencing application (see Section 4). As shown in Figure 15(b), the available noise magnitude is not adequate to induce the probabilistic behavior needed for the probability parameter \( p = 0.85 \). Note that, for this specific example of Figure 15(a), the NSR value is equal to 0.48 independent of the feature size (from Law 3).

For convenience, we shall refer to the amount by which the freely available noise magnitude is less than that dictated by a particular NSR value given a desired probability parameter \( p \)—\( \alpha \) in the case of the above example—as the noise oracle gap, or the noise gap when convenient. Informally, extra energy ought to be expended to bridge this gap (Figure 16), by amplifying the noise magnitude so that the resulting amplified noise is at the adequate magnitude. For example, in 2004, with a nominal operating voltage determined from the ITRS roadmap to
Figure 16. Bridging the gap between available noise and that needed by a particular probability parameter value $p = 0.85$, and the associated amplification energy cost $\beta$ Joules.

Table 6. The energy cost associated with bridging the NSR value based on available thermal noise to that needed to produce a particular probability value $p$ across technology generations.

<table>
<thead>
<tr>
<th>Technology</th>
<th>$p = 0.85$</th>
<th>$p = 0.7$</th>
<th>$p = 0.65$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.25\mu m$</td>
<td>$41$</td>
<td>$61$</td>
<td>$83$</td>
</tr>
<tr>
<td>$0.18\mu m$</td>
<td>$26$</td>
<td>$39$</td>
<td>$53$</td>
</tr>
<tr>
<td>$0.13\mu m$</td>
<td>$17$</td>
<td>$25$</td>
<td>$33$</td>
</tr>
<tr>
<td>$90nm$</td>
<td>$10$</td>
<td>$15$</td>
<td>$83$</td>
</tr>
<tr>
<td>$65nm$</td>
<td>$6.9$</td>
<td>$10.3$</td>
<td>$14$</td>
</tr>
<tr>
<td>$32nm$</td>
<td>$3.1$</td>
<td>$4.5$</td>
<td>$6.2$</td>
</tr>
<tr>
<td>$22nm$</td>
<td>$1.9$</td>
<td>$2.8$</td>
<td>$3.74$</td>
</tr>
</tbody>
</table>

be 1.2V for a 90nm process, and considering the available noise (freely from an oracle) to be 100mV, $\text{NSR}= \frac{0.1}{1.2} = 0.083$; recall that the noise-to-signal ratio is defined to be $\text{NSR}= \frac{\sigma}{V_{dd}}$. However, observe in Figure 15(b) that the NSR needed to achieve a probability parameter $p = 0.85$ in our example, is 0.48 > 0.083. Thus, in our example of nominal $V_{dd} = 1.2V$, the only way in which the NSR value can be raised to the desired value of 0.48 is by amplifying noise from 100mV to 576mV. As a result, the new value of NSR is $\frac{0.576}{1.2} = 0.48$. Therefore, as shown in Figure 16, in the absence of a freely available noise oracle, we use a (subthreshold) amplifier that consumes very low energy, to amplify the available noise.

We enumerate the cost associated with amplifying one bit, which is input to a single inverter switching step (Figure 16), across technology generations of interest to us in Table 6 and Figure 17 for different probability parameters ($p = 0.85$, $p = 0.7$ and $p = 0.65$). Our estimates for amplification cost are based on a sub-threshold amplifier design described in detail by Cheemalavagu, Korkmaz, Palem, Akgul and Chakrapani [8]. Table 6 shows the amplifier gain needed to provide a noise RMS value, given a particular operating $V_{dd}$ value, to achieve a particular probability value of $p$.

In Table 6, we also show the associated energy cost of the amplifier ($E_A$), which is illustrated in Figure 17. Here, the estimated values of energy cost ($E_A$) are determined as follows. These estimates were obtained considering the three scaling factors caused by (i) $V_{dd}$ and denoted by
Figure 17. Amplifier energy cost over technology generations for probability parameter values of ranging from $p = 0.65$ to $p = 0.85$.

$s_1$, which is projected to be $\approx 0.7$ per generation, (ii) the device length denoted by $s_2$, which is projected to be $\approx 0.8$ per generation, and (iii) gain of the amplifier denoted by $s_3$ as dictated by the NSR—see Figure 15 for example. Here, $s_3$ corresponds to the estimated scaling factor of the amplifier’s gain needed to amplify the available noise magnitude; since the operating $V_{dd}$ voltage values scale down across technology generations, the noise magnitudes, and the associated amplification need.equivalently, the gain of the amplifier—would also decrease. Thus, we capture the effect of technology scaling on the amplification need (gain) through $s_3$. Our estimation of the energy consumed by the amplifier is based on the reasoning that the energy cost of the amplifier $E_A$ is proportional to $(s_1^2 \cdot s_2 \cdot s_3)$, where $s_1^2$ corresponds to the quadratic dependence of $E_A$ on the supply voltage, and $s_2$ and $s_3$, respectively, correspond to the linear dependence of $E_A$ to the capacitance value and the gain of the amplifier as they scale. Note that in this report, we estimate the dependence of $E_A$ on the gain of the amplifier to be linear for simplicity.

As seen from Table 6 and graphically from Figure 17, the amplification gain needed to achieve a particular value of $p$ and the associated energy cost of the amplifier decreases dramatically with technology scaling. For example, to realize a value of $p = 0.85$, as we scale from a $0.25\mu m$ process to a $22nm$ process, which results in the associated energy cost to be reduced from $340fJ$ to $0.24fJ$, the corresponding gain of the amplifier is reduced from 41 to 1.9. Note that, the results shown in Table 6 and Figure 17 are based on the fact that the available noise is supplied from a resistor, providing a thermal noise RMS value in the range of 10 millivolts to 20 millivolts (using a resistor of 10 G$\Omega$ exposed to a frequency bandwidth between 1MHz to 10MHz). However, in actual process technology, especially below $50nm$, the “available noise”—derived from crosstalk noise, substrate noise, thermal agitation, power supply noise, and others—is expected to be in much higher magnitudes; therefore, our estimates in Table 6 and Figure 17 should be viewed as conservative in the sense that they allow us to estimate the maximum amount of amplification needed. This follows from the fact that as the magnitude of available noise is increased, to achieve a particular value of $p$, the amount of amplification needed decreases correspondingly.
Figure 18. Reconciled energy-probability relationship in absence of a noise oracle for TSMC 0.25μm process. Noise RMS is 0.4V and associated energy cost of the amplifier is 240fJ.

3.1. Relating the Cost of Noise Amplification to Oracle Based Behaviors

In the earlier sections of this paper, the basic laws of PCMs were characterized based on the premise of noise being available in potentially unbounded magnitudes without incurring any energy and time costs for its availability; this idealized notion was characterized through the notion of a noise oracle O. From this idealization, we subsequently discussed the need for potentially amplifying the noise, as a basis for realizing the desired NSR value. In so doing, we introduced an ultra low-energy (sub-threshold) noise amplifier and estimated its energy cost as technology scales down to 22nm. In this section, we will briefly reinterpret this cost associated with amplification, within the context of the oracle O and state an extended law of PCMs when the oracle has an associated explicit cost.

To illustrate this issue, we consider the TSMC 0.25μm technology and probabilities in the range of \( p = 0.7 \) to \( p = 1 \). The effects on the energy consumed by each switching step are shown in Figure 18, both in the context of an available noise oracle, as well as in the context where the noise amplitude needs to be amplified and thus, the cost associated with the oracle is explicitly accounted. As shown in this figure, the energy cost of amplification is directly additive, and is 240fJ with a noise magnitude of \( \sigma = 0.4V \). We also recall from the discussion that clarified the lowering of this amplification cost with technology scaling (Figure 17) and thus, in absolute terms, the separation between the two trends due to amplification cost illustrated in Figure 18 will decrease with smaller CMOS feature sizes.

With this as background, we will conclude the basic characterization of the laws of PCMs, by explicitly accounting for the cost for amplification \( E_A \) as follows. In the absence of a noise oracle, the total energy \( (E_{TOT}) \) consumed by a probabilistic inverter during a switching step, for a fixed value of \( p \) and \( \sigma \), is the sum of the energy consumed by the amplifier \( (E_A) \) and that consumed by the inverter \( (E) \).

While the amplification cost decreases across technology generations, the inverter energy consumption also decreases; however, the decrease in amplifier energy is significantly more rapid than the decrease in inverter energy. This is due to the fact that as \( V_{dd} \) scales down over technology generations, the required noise magnitude \( (\sigma) \) also decreases—according to Law 3—and so does the amplification cost. Based on the estimates from Table 6, we show the corresponding fraction of the amplifier energy and the fraction of the inverter energy across technology
<table>
<thead>
<tr>
<th>Technology</th>
<th>Amplifier energy %</th>
<th>Inverter energy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25μm</td>
<td>94</td>
<td>6</td>
</tr>
<tr>
<td>0.18μm</td>
<td>93</td>
<td>7</td>
</tr>
<tr>
<td>0.13μm</td>
<td>92</td>
<td>8</td>
</tr>
<tr>
<td>90nm</td>
<td>91</td>
<td>9</td>
</tr>
<tr>
<td>65nm</td>
<td>86</td>
<td>14</td>
</tr>
<tr>
<td>32nm</td>
<td>81</td>
<td>19</td>
</tr>
<tr>
<td>22nm</td>
<td>74</td>
<td>26</td>
</tr>
</tbody>
</table>

\[ p = 0.85 \]

| Table 7. | The estimated % fractions of amplifier energy and inverter energy to the total energy consumption for \( p = 0.85 \) across technology generations. |

generations in Table 7 for \( p = 0.85 \). As seen from the table, the fraction of amplification cost decreases with scaling; for example, the energy consumption due to amplifier constitutes 94% of the total energy of a probabilistic inverter designed in 0.25μm process and it constitutes 74% of the total energy of a probabilistic inverter designed in 22nm process.

4. Realizing PCMOS Based Low-energy Architectures

In order to better understand the energy and performance benefits that can be derived at the architectural level based on PCMOS technology, we propose a system-on-a-chip (SoC) architecture as shown in Figure 19. PCMOS serves to realize a co-processor, and the probabilistic content of the computation is executed on this co-processor. Thus, early yet significant adoption of PCMOS is envisioned by us to be application specific, and evolving to a context that is domain specific. As shown in the figure, the low-energy host processor will be used to compute all of the deterministic components of the application. A typical host processor will be a low-energy StrongArm \[ \text{[45]} \], a MIPS \[ \text{[41]} \] or an equivalent low-energy embedded processor, coupled to the co-processor via the system bus. Thus, the host processor accesses the co-processor through memory mapped I/O. The host could also be a custom-fit processor in its own right; for details about the concept of a SOC and further details about custom-fit processors, please see Lyonnard et al. \[ \text{[21]} \], Tensilica \[ \text{[50]} \] and Wang et al. \[ \text{[48]} \]. Thus, in this paper, we also consider a host designed as a custom ASIC and analyze the efficiency of a host as a factor that leverages the benefits of PCMOS (see Section 4.2). The applications that we initially considered are the widely-used Bayesian inferencing using Bayesian networks (BN) \[ \text{[39, 34]} \], vector covering problem using randomized neural networks (RNN) \[ \text{[14]} \], a classification application using probabilistic cellular automata (PCA) \[ \text{[12]} \], and an hyper-encryption application (HE) \[ \text{[10]} \]. The domains of interest that we intend to explore next, include signal and image processing, cryptography and security, as well as controls.

The two basic criteria of interest in realizing efficient application-specific SOC architectures are the performance (typically the running-time of the application) and its energy consumption (or its derivative power). Thus, our goal shall be to realize architectures that are significantly more efficient than a conventional processor using both of these criteria. Thus, our first metric for consideration shall be the energy-performance product of an architecture of a particular application—akin to the energy-delay product in circuit design—defined below.
**Figure 19.** A “SoC-like” architecture for realizing low-energy computing architectures using PCMOS.

**Energy-Performance Product (EPP).** EPP is defined as the product of the application level energy (measured in Joules) and performance (measured in number of cycles).

EPP will be used as the chief metric of interest to evaluate various implementations. Given the EPP of two alternate implementations—for example, the case when the entire algorithm is implemented as software executing on the host referred to as the *baseline*, compared to the case where the deterministic part of the algorithm is executed on the host with the probabilistic part executing on a PCMOS co-processor—they can be compared by computing the ratio of their individual EPP values.

Since our goal is to compare the energy and performance gains realized through using PCMOS technology, we refine this notion and define the metric: EPP gain, which is denoted as \( \Gamma \), and defined as follows.

**EPP Gain (\( \Gamma \)).** EPP gain, denoted as \( \Gamma \), is the ratio of the EPP of the baseline to the EPP of a particular implementation. The EPP gain of a particular implementation \( \mathcal{I} \) is determined as

\[
\Gamma_{\mathcal{I}} = \frac{\text{Energy}_B \times \text{Time}_B}{\text{Energy}_I \times \text{Time}_I}
\]

In Equation 5, *baseline* denoted as \( B \) refers to the case when the entire application is realized using the host (for example, a StrongArm SA-1100 processor or equivalently, an ASIC realization of deterministic content) only, *without recourse to a co-processor*. For example, in the context of a Bayesian network application, the well-known Junction Tree (deterministic) algorithm [23] shall serve as the baseline. Thus, the numerator of \( \Gamma_{\mathcal{I}} \) is derived by considering the entire application to be executing deterministically on the host only. The corresponding architecture realization, which we refer to as the baseline, is shown in Figure 20(a). While the baseline and hence the numerator of the EPP gain metric thus far has been a purely deterministic realization of a deterministic algorithm corresponding to the case shown in Figure 20(a), in the context of applications that do not have a (purely) deterministic application context such as RNN for example, the software based emulation from Figure 20(b) shall serve as the baseline. We will also adopt this approach whenever the deterministic realizations do not exist or are impractically inefficient.

Alternate scenarios or “mixes” of partitioning the application across the host and co-processor
are shown in Figure 20 as cases (b), (c) and (d). In the scenario wherein the target computational platform is constituted exclusively of a host without any co-processor—our baseline case—and when considering a probabilistic variant of the Bayesian network for example, the probabilistic component is “emulated” using pseudo-random bit generation in software (as shown in Figure 20(b)).

A further refinement of this approach is to consider a co-processor (Figure 20(c)) wherein the pseudo-random emulation is realized using a customized co-processor—typically a pseudo-random number generator (PRNG, see Park and Miller [33]), with the corresponding $\Gamma$ value again determined against a deterministic baseline. Finally, as shown in Figure 20(d), the co-processor and hence the probabilistic computational component is realized using PCMOS.

These cases (shown in Figure 20(b), (c), and (d)) ought to constitute a progression of increasing improvements quantified by the corresponding $\Gamma$ values, starting with software based (Figure 20(b)) and custom-hardware based (Figure 20(c)) pseudo-probabilistic emulation of probabilistic applications, to authentic probabilistic computations realized using PCMOS with a StrongArm SA-1100 host (Figure 20(d)).

Besides the design choices shown in Figure 20, we also consider two alternative implementations of the host processor computing the deterministic content of the application. The two alternatives for the host are a StrongArm SA-1100 processor shown in Figure 21(a) and a custom ASIC shown in Figure 21(b). We will see in Section 4.2 that the PCMOS gains are more significant if a custom ASIC realization of the host is used.

In Section 4.1, we will summarize the probabilistic algorithms and the associated applications, and in Section 4.2, we will quantify PCMOS savings using metrics introduced above, in the context of the applications from Section 4.1.

4.1. The Suite of Applications

Probabilistic algorithms, in computer science parlance, “toss coins” i.e., consume bits from a source of randomness. Examples of such algorithms include the celebrated test for primality, referred to as the Rabin-Strassen-Solovay algorithm [37, 42], used as a key building block in RSA public-key cryptosystems. To demonstrate the utility and efficacy of a PSOC we consider
applications that employ probabilistic algorithms that implement Bayesian networks (BN) [39, 34], random neural networks (RNN) [14], probabilistic cellular automata (PCA) [12] and hyper-encryption (HE) [10]. Probabilistic content and utility in a wide range of application scenarios are the common characteristics of these algorithms.

Among these algorithm kernels that we have implemented, the Bayesian networks are used in applications such as spam-filters, cognitive learning, battlefield planning [35], windows printer trouble shooting and hospital patient management [3]. Random neural networks are used in image and pattern classification, network routing and optimization of NP-hard problems such as finding the vertex cover of a graph. Probabilistic cellular automata are used in pattern or string classification [12], and finally, hyper-encryption is used in security applications for message encryption.

4.2. PCMOS Gains for Application Specific SOC Architectures

In this section, we summarize the resulting gains of PCMOS for all the applications from Section 4.1. PCMOS gains illustrated in Table 8 are at the scope of an application ranging from a factor of (about) one for the HE application, to a factor of 300 in the context of the RNN application. As seen in Table 8, these gains refer to the EPP gain of PCMOS over the baseline. In the case of the BN application, the baseline is a deterministic (junction-tree) algorithm running on the StrongArm SA-1100 processor, which refers to the case shown in Figure 20(a). In the case of the other applications, the baseline is a probabilistic algorithm running on the StrongArm SA-1100 processor, which refers to the case shown in Figure 20(b). A range of EPP gains are observed whenever multiple data points are available, for example, in the context of BN where different data points correspond to different networks, the probabilistic content of the application also varies, the corresponding gain increases from a factor of 12.5 to an impressive factor of 291 due to increase in the probabilistic proportion of the application content. Similar increases are observed for the other applications as well, caused by an increase in the proportion of their probabilistic contents.

One observation is that—besides probabilistic content—the EPP gain also depends on the efficiency of the host serving as the baseline. If the energy consumed on the host to compute the deterministic part of an application is more dominant than the energy consumed to compute the probabilistic part on the co-processor, then the EPP gain would be very small. This fact is observed in the case of the HE application, where the host SA-1100 energy is dominant and hence the resulting EPP ratio is only 1.12 (see Table 8). Moreover, as the host processor’s
<table>
<thead>
<tr>
<th>Application</th>
<th>Baseline</th>
<th>EPP Gain of PCMOS = $\Gamma_{PCMOS}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Min</td>
</tr>
<tr>
<td>BN</td>
<td>Deterministic junction-tree alg. on StrongArm (Figure 20(a))</td>
<td>12.5</td>
</tr>
<tr>
<td>RNN</td>
<td>Probabilistic alg. on StrongArm (Figure 20(b))</td>
<td>226</td>
</tr>
<tr>
<td>PCA</td>
<td>on StrongArm</td>
<td>61</td>
</tr>
<tr>
<td>HE</td>
<td>(Figure 20(b))</td>
<td>1.12</td>
</tr>
</tbody>
</table>

**Table 8.** Application level min and max EPP gains of PCMOS over the baseline implementation, where the StrongARM SA-1100 processor serves as the host.

Energy increases significantly relative to that of the co-processor, alternate realizations of the co-processor such as variations from CMOS to PCMOS cannot be differentiated through the $\Gamma$ metric.

Therefore, we also consider the effect of host efficiency on the EPP gains of PCMOS through alternate implementations of the host. The particular two alternatives that we have studied are the StrongArm SA-1100 processor and the full custom ASIC realizations of the host. Table 9 shows the EPP gains of PCMOS where the baseline is taken to be the CMOS realization of the co-processor in the case of the two applications that we have realized, namely, the hyper-encryption and the probabilistic cellular automata. As seen from the table, the gain of PCMOS over CMOS is negligible (around 1) when the host is StrongArm for both applications. By contrast, when the StrongArm host is replaced by its much more efficient ASIC counterpart, the gains increase significantly, from 1.06 to 9.48 in the case of hyper-encryption and from 1.1 to 1969.7 in the case of the probabilistic cellular automata.

<table>
<thead>
<tr>
<th>Application</th>
<th>EPP gain of PCMOS = $\Gamma_{PCMOS}$ when Baseline=CMOS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Host: StrongArm SA-1100</td>
</tr>
<tr>
<td>Hyper-Encryption</td>
<td>1.06</td>
</tr>
<tr>
<td>Probabilistic Cellular Automata</td>
<td>1.1</td>
</tr>
</tbody>
</table>

**Table 9.** Comparing the EPP gains for the HE and PCA applications when host is the StrongArm SA-1100 processor and when it is custom ASIC.

5. Quality of Probabilistic Bits

The _quality_ of randomization also plays an important role in many probabilistic algorithms. Among these algorithms that are sensitive to quality of randomness, Monte Carlo simulations, for example, have been shown to yield incorrect results when poor quality pseudo-random number generators are used as the source of random bits [11]. In addition, the strength of encryption schemes like Hyper-Encryption can be severely compromised unless true-random (as opposed to pseudo-random) sequences are provided. For such applications whose correctness, and hence utility, depend on the quality of random bits, it is important to compare the quality of randomization afforded by PCMOS and conventional CMOS based implementations.
In Figure 22, we show the statistical tests from the NIST suite [38] applied to compare and evaluate the quality of random bit sequences generated by a PC莫斯 inverter (switch) and those generated by a CMOS-based pseudo-random number generator; for both cases, \( p \) is considered to be 0.5. The random sequences in the case of PCMOS have been produced from the actual chip measurements of a PC莫斯 inverter from 0.25\( \mu \)m TSMC prototype and those of CMOS from HSpice simulations of the hardware implementation of the Park-Miller [33] random number generator.

Among these tests and to highlight a few, the frequency tests evaluate the frequency of 0s and 1s—whether the fraction of 1s to 0s is close to 0.5. The runs test determine contiguous sequence of 1s in a block. The rank test checks the linear dependence, while the FFT and approximate entropy tests detect periodicity and frequency of overlapping patterns. The template matching tests detect repetitions of non-periodic patterns and the universal statistical test as well as the lempel-ziv test detect whether the random sequence can be compressed. The tests are run on random bit sequences of length 20,000,000. In evaluating the test results, we use the same testing strategy and criteria as recommended by the NIST suite. Specifically, the test results shown in parenthesis in the table are compared against a threshold (which is 0.93) used to determine whether the sequence passes or fails a test. The result indicates the proportion of subsequences (tested through iterations) that pass from the random sequence being tested. As seen from the figure, PCMOS passes 11 tests out of 14 whereas CMOS passes only 7 of these tests.

### 6. PCMOS and Parameter Variations

In Section 2, we introduced the basic laws governing the behavior of PCMOS. We recall from analytical models governing the behavior of the probabilistic switches introduced there (Figure 3) that the various switches were identical in their behavior, as was the statistic characterization of the noise inducing the probabilistic behavior. Thus, there are no observable variations “across” switches, whereas the results of switching were potentially different over time and hence across successive switching steps. We show this behavior in Figure 23 in the context of two switches sw1 and sw2.

To reiterate, the type of time-varying probabilistic switching indicated above is caused by noise, whose impact on CMOS as devices scale into the nanoscale yields probabilistic behaviors—for convenience, we shall refer to this type of PCMOS as TPCMOS. By contrast, probabilistic behaviors are also anticipated in future CMOS technologies, induced by an independent and quite different source—parameter variations [2, 4, 5, 27]. In this context, as illustrated in Figure 24, each individual device (or switch), once fabricated, behaves in an identical manner on each
Figure 23. *PC*MO*S* switches that are time-varying wherein switches SW1 and SW2 have identical deterministic and (additive) noise characteristics following the model from Figure 3.

Figure 24. Variations of device (switch) characteristics across the surface (space) of an integrated circuit.

successive step. However, as feature sizes become smaller, devices start behaving very different “across” the surface of a single integrated circuit—we shall refer to this type of *PC*MO*S* as space-varying, or *SP*CMOS for short, since the variations are in (physical) space. However, as the feature sizes of these devices (or switches) scale down, the number of such switches packaged as a single integrated circuit start approaching Meindl’s Tera-scale regime [24]. As a result of these increasingly large number of devices or switches in a single integrated circuit, these variations can be modeled statistically in a meaningful manner and attempts to develop such model are being undertaken by several research groups (see Bowman et. al [5], Nassif [27] and Azizi et. al [2] as examples). Two questions suggest themselves immediately within the context of such space-varying PC*MO*S, or *SP*CMOS for short.

1. The first and obvious direction to consider is an architectural approach based on *SP*CMOS, similar to that outlined in Section 4 in the context of TPC*MO*S. The goal here will be to
Figure 25. (a) Probabilistic PCMOSS-based hardware primitive and associated error control and primary inputs. (b) Chernoff bounds.

take advantage of spatial parameter variations as a resource, much as noise served that purpose to realize ultra-efficient co-processors realizing probabilistic workloads.

2. The next item to consider is perhaps the “holy grail” of the semiconductor industry for realizing reliable information from unreliable computing elements—the goal being to continue exploiting the benefits of Moore’s law in the context of general purpose microprocessors, memory and peripherals supporting deterministic applications.

In this context, the concept illustrated in Figure 25(a) serves to illustrate an approach that is suitable, within the context of computing reliably using unreliable switches based on TPCMOSS or SPCMOSS. The basic concept involves integrating “error correction” denoted by bits $e_0, e_1, \ldots, e_{k'}$ in addition to the primary input bits $i_0, i_1, \ldots, i_k$. The basic building block in question that we are considering is drawn from the digital signal processing domain and is a finite impulse response (FIR) filter [22, 49]. The goal of our design is to minimize the energy-performance product as before, which entails minimizing the overhead due to error correction, characterized by $k'$ in a preliminary way.

As a first step, we will consider threshold voltage ($V_{th}$) to be our parameter and explore its variations. Therefore, our methodology in implementing error correction mechanism will be aimed at controlling the variations in $V_{th}$ as it varies across devices. We propose to cope with $V_{th}$ variations (through the error-correction mechanism) by varying the bulk (substrate) voltages of the transistors. We intend to characterize the relation between the probability of correctness $p$ and variations in $V_{th}$. One analytical method that we will use is the application of Chernoff bounds [1] to characterize and control $V_{th}$ variations, and hence the probability parameter $p$ of an SPCMOSS device. Specifically, the Chernoff bound is a simple estimate of the area under the tails of a distribution, and is thus a crucial tool in bounding error precisely. As seen from Figure 25(b), the Chernoff bound implies that a small deviation ($\delta$) from the mean ($\mu$) causes a large change in the probability $p$, due to the exponential dependence between $p$ and the deviation. In the context of the proposed effort, this bound will be utilized to determine sharp boundaries for $p$, as they relate to the variations in $V_{th}$.

Both of these pursuits are currently being investigated and are expected to be reported in an update of this report during the Spring of 2006.
7. Remarks

Computing, from the early notion of a stored-program computer [47], through its enormous impact starting with the concept of an integrated circuit [18, 36] to the modern microprocessor [16] has been viewed as a vehicle for computing deterministic algorithms—in this scenario, the answers are deemed to be always correct. Purely at the algorithmic and concomitant theoretical levels, this penchant for determinism was replaced with considerable success by relying on the counterintuitive notion of algorithms, one of whose design metrics was the “error” associated with the algorithm—referred to popularly as probabilistic or randomized algorithms, they represent a class of algorithms wherein, the designer, knowingly expects the algorithm to be erroneous in a well-understood manner quantified by an associated probability. Led by the ground-breaking work embodied in the algorithm for primality testing, often referred to as the Rabin-Solovay-Strassen algorithm (see Rabin [37, 42]), the goal was to accelerate the algorithm’s execution time, in return for losing—often a disproportionately small amount—a degree of correctness in the algorithm’s behavior.

While an algorithm’s (or equivalently, its instantiation as a program’s) execution time has been the overwhelming consideration in the design of modern computing systems, the energy consumed by computing platforms, the associated heat dissipation as well hurdles to continued device scaling in the (low) nano-meter scale have all emerged and equally, if not more important considerations in the innovation of computing switches, circuits and architectures. In a novel approach to overcome these challenges, Palem [28, 29, 31] outlined a framework for realizing computations based on models and devices that are known to be erroneous—albeit in a well-understood manner. A natural consequence of this work was the need to model, characterize and measure the behaviors of CMOS devices and their derived switches, wherein the associated probability of error is a “design parameter”—especially since traditional characterizations including the ITRS roadmap do not explicitly address the probability of error, or equivalently the probability of its correct execution quantified through the parameter $p$ in this work. Consequently and over the past two years, our group has undertaken an effort to explore, characterize, and exploit the probability of a CMOS based computing switch being correct. The results of such modeling and characterization were elucidated in this report, yielding an extension to the roadmap through a novel type of computing device that we refer to as PCMOS.

Phenomenologically, the challenges posed to CMOS can be broadly characterized into two categories. First, as device feature sizes and the related operating voltages diminish in magnitude in keeping with Moore’s law, noise manifested through a range of sources, is viewed as an impediment to their correct operation. As detailed in this report, this phenomenon yields switches producing outputs that are probabilistically characterized on each successive step, over time, whereas the individual devices are indistinguishable such that in the absence of noise, they have identical deterministic characteristics. In Section 6, we referred to this type of a probabilistic variation as time-varying PCMOS or TPCMOS for short. In Section 4 and in two recently announced results [29, 7, 32], we have shown that through a novel PSOC architectural framework, orders of magnitude gains to the energy consumed as well as to the performance (execution time) can be achieved in the context of interesting and various probabilistic workloads and applications. Further exploration of TPCMOS as a novel alternative to realizing ultra low-energy architectures through the PSOC framework, for increasingly probabilistic workloads such as speech, vision, voice, security and a range of other emerging applications is a very promising direction for further work.

A dual phenomenon, also often associated in the context of CMOS circuits and switches with the phrase “probabilistic devices” is caused by parameter variations. Here and in contrast to
the TPCMOS concept, once fabricated, an individual switch behaves in the same manner across successive switching steps and hence in time. However, large and statistically characterized variations are observed to occur across devices in a Tera-scale IC, and this trend in variations is expected to increase. Since the probabilistic variations are across devices, it is convenient to characterize this concept as a spatial variation that we refer to as space-varying PC莫斯 or SPC莫斯. A detailed characterization of the characteristics of SPC莫斯 based switches, akin to that presented in this report in the context of TPCMOS, will be the subject of a companion report to be released over the next few months. Additionally, realizing PSOC architectures that are ultra energy-efficient and thus can provide computing substrates in the context of probabilistic applications using SPC莫斯 is a topic of current exploration by our group.

Moving to the third and final line of enquiry that is at the forefront of our considerations, based on the detailed characterizations embodied in the laws of PC莫斯 detailed in this report, we are actively pursuing an approach to develop architectural building blocks—the DSP domain of considerable initial interest where the switch’s error manifests itself as the signal-to-noise ratio or alternately as distortion—wherein the error germane to a TPCMOS switch can be varied and potentially lowered through the use of error correction techniques augmenting the basic PC莫斯-based switches and circuits. This approach, we believe, is unique in achieving the goal of realizing reliable (or deterministic) computing from unreliable (TPCMOS and SPC莫斯) devices, switches and circuits.
References


