Probabilistic CMOS Technology: A Survey and Future Directions

Bilge E. S. Akgul, Lakshmi N. Chakrapani, Pinar Korkmaz and Krishna V. Palem
Center for Research on Embedded Systems and Technology
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332–0250, USA.

Abstract— Highly scaled CMOS devices in the nanoscale regime would inevitably exhibit statistical or probabilistic behavior. Such behavior is due to process variations and other perturbations such as noise. Therefore current circuit design methodologies, which depend on the existence of deterministic and uniform devices with no consideration for either power consumption or probabilistic behavior, would no longer be sufficient to design robust circuits. To help overcome this challenge, we have been characterizing CMOS devices with probabilistic behavior (probabilistic CMOS or PCMOS devices) at several levels: from foundational principles to analytical modeling, simulation, fabrication and measurement, as well as innovative approaches to harnessing PCMOS devices in system-on-a-chip architectures which can implement a wide range of applications. In this paper, we present a broad overview of our contributions in the domain of PCMOS, and outline ongoing work and future challenges in this area.

I. INTRODUCTION AND OVERVIEW

Sustained device scaling into the nanometer regime faces several hurdles. Manufacturing difficulties yield devices with parameter variations and since these devices are likely to operate close to the thermal limit, they are susceptible to perturbations due to noise [1], [2], [3]. We refer to such CMOS devices whose behavior is probabilistic as *probabilistic* CMOS devices, or PCMOS devices for short. Clearly, current circuit design methodologies are inadequate to design robust circuits in the presence of these perturbations, since they depend on the devices with deterministic behavior. To design robust circuits and architecture in the presence of this (inevitable) statistical behavior at the device level, it has been speculated that a shift in the design paradigm—from the current day deterministic designs to statistical or *probabilistic designs* of the future— would be necessary [4].

We have addressed this issue of probabilistic design at several levels, from foundational models [5], [6] to practical system-on-a-chip architectures which leverage PCMOS technology for applications from the cognitive and embedded domains [7]. This paper presents a broad overview of our contributions in the area of probabilistic CMOS; several of these results have appeared in prior publications [7], [8], [9], [5], [6]. In this paper we survey our past work, present some new results—in particular a formulation of the two laws of PCMOS using asymptotic notation as well as a scheme for

efficient implementation of PCMOS based system on a chip architectures.

The rest of the paper is organized as follows. In Section II we outline the foundational principles of PCMOS technology based on the *probabilistic switch*. In Section III we show how these abstract foundational models can be realized in the domain of CMOS in the form of noise susceptible scaled CMOS devices operating at low voltages and state the two laws of PCMOS technology using novel asymptotic notions. For our present purpose, it is convenient to partition the application domain into three groups (i) applications which benefit from (or harness) probabilistic behavior at the device level naturally, (ii) applications that can tolerate (and trade off) probabilistic behavior at the device level (but do not need such behavior naturally) and (iii) applications which cannot tolerate probabilistic behavior at all. We will briefly sketch our approach towards implementing PCMOS based architectures, in Section IV. In Section V we outline other challenges such as design for manufacturability, and through a probabilistic approach present a novel approach towards addressing one such problem—the problem of multiple voltage levels on a chip. Finally in Section VI we conclude and sketch future directions of inquiry.

II. FOUNDATIONAL PRINCIPLES

We have innovated probabilistic switches as foundational models which incorporate probabilistic behavior as well as energy consumption as first class citizens [6]. A probabilistic switch is a switch, which realizes a *probabilistic one-bit switching function*. As illustrated in Figure 1, the four deterministic one bit switching functions (Figure 1(a)) have a probabilistic counterpart (Figure 1(b)) with an *explicit* probability parameter (probability of correctness) p. Of these, the two constant functions are trivial and the others are non-trivial. We consider an abstract probabilistic switch sw to be the one which realizes one of these four probabilistic switching functions. Such elementary probabilistic switches may be composed to realize primitive boolean functions, such as AND, OR, NOT functions.

Principles of statistical thermodynamics may be applied to such switches to quantify their energy consumption, and hence the energy consumption (or energy complexity) of a network

Input	output
0	0
1	1

Identity Function

Input	output
0	1
1	0

Complement Function

Input	output	
0	0 (p)	1 (1-p)
1	1 (p)	0 (1-p)

Identity Function

Input	output	
0	1 (p)	0 (1-p)
1	0 (p)	1 (1-p)

Complement Function

Input	output
0	0
1	0

Constant Function

Input	output
0	1
1	1

Constant Function

Input	output	
0	0 (p)	1 (1-p)
1	0 (p)	1 (1-p)

Constant Function

Input	output	
0	1 (p)	0 (1-p)
1	1 (p)	0 (1-p)

Constant Function

(a)   (b)

Fig. 1. (a) Deterministic one bit switching functions (b) Their probabilistic counterparts with probability parameter (probability of correctness) p
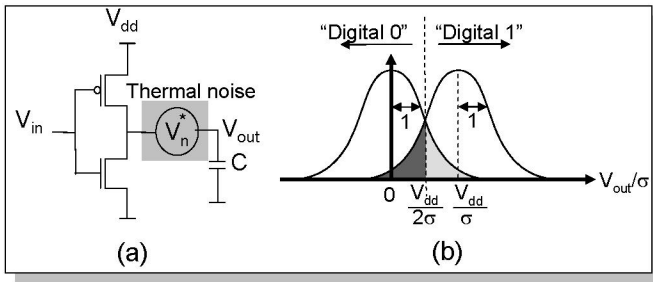

Fig. 2. (a) PCMOS switch (b) Representation of digital values 0 and 1 and the probability of error for a PCMOS switch

of such switches. While a switch that realizes the deterministic non-trivial switching function consumes at least $\kappa t \ln 2$ Joules of energy [10], a probabilistic switch can realize a probabilistic non-trivial switching function with $\kappa t \ln(2p)$ Joules of energy where p is the probability parameter [6]. For a complete definition of a probabilistic switch, the operation of a network of probabilistic switches and a derivation of energy complexity of such networks, the reader is referred to Palem [6].

III. THE CMOS DOMAIN: PROBABILISTIC CMOS

We have demonstrated how abstract probabilistic switches might be used to implement useful logic, where the probability of correct operation p is a parameter. Probabilistic switches serve as a foundational model for physical realizations of highly scaled devices as well as emerging non-CMOS devices. In the domain of CMOS, they model noise-susceptible CMOS (or PCMOS) devices operating at very low voltages [9]. To show that PCMOS based realizations correspond to abstract probabilistic switches, we have demonstrated two key characteristics of PCMOS: (i) Probabilistic switching and (ii) Energy savings through probabilistic switching. These characteristics were demonstrated through analytical modeling and HSpice based simulations [9], [11] as well as actual measurements of fabricated PCMOS based devices. We will now formalize these behavioral and energy characteristics of PCMOS switches using asymptotic notions from computer science [12], [13], [14] in the form of two laws.

For a PCMOS switch as shown in Figure 2 (a), the output voltage (V_{out}) is probabilistic due to (thermal) noise coupled to its output. This noise has a mean value of 0 and a variance of

σ^2. The normalized output voltage $\frac{V_{out}}{\sigma}$ can be represented as a random variable having a Gaussian distribution as shown in Figure 2 (b), and the variance of the distribution is 1. The mean value of the distribution is 0 if the correct output is supposed to be a digital 0, and $\frac{V_{dd}}{\sigma}$ if the correct output is supposed to be a digital 1. In this representation, the two shaded regions of Figure 2 (b) (which are equal in area) correspond to the probability of error per switching of a PCMOS inverter. From this, we determine the probability of correctness denoted as p, by computing the area in the shaded regions and express p as

$$p = 1 - \frac{1}{2}erfc\left(\frac{V_{dd}}{2\sqrt{2}\sigma}\right) \qquad (1)$$

where $erfc(x)$ is the complementary error function

$$erfc(x) = \frac{2}{\sqrt{\pi}}\int_x^\infty e^{-t^2}dt \qquad (2)$$

Using the bounds for $erfc$ derived by Ermolova and Haggman [15], we have

$$p < 1 - 0.28e^{-1.275\frac{V_{dd}^2}{8\sigma^2}} \qquad (3)$$

Using this expression to lower-bound V_{dd} and hence the switching energy $\frac{1}{2}CV_{dd}^2$, we have, for a given value of p,

$$E(p,C,\sigma) > C\sigma^2\left(\frac{4}{1.275}\right)\ln\left(\frac{0.28}{1-p}\right) \qquad (4)$$

Clearly, E is a function of the capacitance C, determined by the technology generation, σ the root-mean-square (RMS) value of the noise and the probability of correctness p. For a fixed value of $C = \hat{C}$ and $p = \hat{p}$, $\tilde{E}_{\hat{C},\hat{p}}(\sigma)$ is defined as $\tilde{E}_{\hat{C},\hat{p}}(\sigma) = \hat{C}\sigma^2\left(\frac{4}{1.275}\right)\ln\left(\frac{0.28}{1-\hat{p}}\right)$. Similarly for fixed values of $C = \hat{C}$ and $\sigma = \hat{\sigma}$, $\hat{E}_{\hat{C},\hat{\sigma}}$ a function of p, is defined as $\hat{E}_{\hat{C},\hat{\sigma}}(p) = \hat{C}\hat{\sigma}^2\left(\frac{4}{1.275}\right)\ln\left(\frac{0.28}{1-p}\right)$.

In computer science, the notion of asymptotic complexity is widely used to study the efficiency of algorithms. Usually, efficiency is characterized by the growth of the running time (or space), of the algorithm as a function of the size of its inputs [12], [13], [14]. The O notation provides an asymptotic *upper-bound*. In this context, for a function $f(x)$ where x is from the set of natural numbers

$$f(x) = O\left(h(x)\right)$$

Given any function $h(x)$, whenever there exists positive constants c, x_0 such that $\forall x \geq x_0, 0 \leq f(x) \leq ch(x)$.

Similarly, the symbol Ω is used to characterize an asymptotic *lower-bound* on the rate of growth of a function. For a function $f(x)$ as before,

$$f(x) = \Omega\left(h(x)\right)$$

2

whenever there exists positive constants c, x_0 such that $\forall x \geq x_0, 0 \leq ch(x) \leq f(x)$. In this context, the O and the Ω notation is defined for functions over the domain of natural numbers. We now extend this notion to the domain of reals. For any $y \in (\alpha, \beta)$ where $\alpha, \beta \in \{\Re^+ \cup 0\}$

$$\hat{h}(y) = \Omega_r(g(y))$$

whenever there exists a $\gamma \in (\alpha, \beta)$ such that $\forall y \geq \gamma, 0 \leq g(y) \leq \hat{h}(y)$. Intuitively, the conventional asymptotic notations capture the behavior of a function $h(x)$ "for very large" x. Our modified notion Ω_r captures the behavior of a function $\hat{h}(y)$, defined in an interval (α, β). In this case, $\hat{h}(y) = \Omega_r(g(y))$ if there exists some point γ in the interval (α, β) beyond which $0 \leq g(y) \leq \hat{h}(y)$. This notion means "the function $\hat{h}(y)$ eventually *dominates* $g(y)$ in the interval (α, β)". In this paper, we will use this asymptotic approach to determine the rate of growth of energy described in Equation 4, as follows.

Let us now express the lower-bound we stated in (4) using the novel asymptotic (Ω_r) notation. Again, fixing $C = \hat{C}$ and $\sigma = \hat{\sigma}$, let us consider the expression $\hat{C}\hat{\sigma}^2 \left(\frac{4}{1.275}\right) \ln\left(\frac{0.28}{1-p}\right)$ from Equation 4, and compare it against the *exponential* (in p) *function*, $E^e_{\hat{C}, \hat{\sigma}}(p) = \hat{C}\hat{\sigma}^2 e^p$. We note that, when $p = 0.5$,

$$\hat{C}\hat{\sigma}^2 \left(\frac{4}{1.275}\right) \ln\left(\frac{0.28}{1-p}\right) < E^e_{\hat{C}, \hat{\sigma}}(p)$$

Furthermore, both functions are monotone increasing in p and they have equal values at $p \approx 0.87$. Hence,

$$\hat{C}\hat{\sigma}^2 \left(\frac{4}{1.275}\right) \ln\left(\frac{0.28}{1-p}\right) > E^e_{\hat{C}, \hat{\sigma}}(p)$$

whenever $p > 0.87$. Then, from the definition of Ω_r, an asymptotic lower-bound for $\hat{E}_{\hat{C}, \hat{\sigma}}(p)$ in the interval $(0.5, 1)$ is

$$\hat{E}_{\hat{C}, \hat{\sigma}}(p) = \Omega_r(E^e_{\hat{C}, \hat{\sigma}}(p)) \tag{5}$$

Let $E^q_{\hat{C}, \hat{p}}(\sigma) = \hat{C}\left(\frac{4}{1.275}\right) \ln\left(\frac{0.28}{1-\hat{p}}\right) \sigma^2$. Referring to (4) and considering $\tilde{E}_{\hat{C}, \hat{p}}(\sigma)$ for a fixed value of $C = \hat{C}$ and $p = \hat{p}$, using the Ω_r notation, an asymptotic lower-bound for $\tilde{E}_{\hat{C}, \hat{p}}(\sigma)$ is

$$\tilde{E}_{\hat{C}, \hat{p}}(\sigma) = \Omega_r\left(E^q_{\hat{C}, \hat{p}}(\sigma)\right) \tag{6}$$

Observation 1: Whereas the function $E^e_{\hat{C}, \hat{\sigma}}(p)$ grows exponentially in p, for a fixed $C = \hat{C}$ and $\sigma = \hat{\sigma}$, the function $E^q_{\hat{C}, \hat{p}}(\sigma)$ grows quadratically in σ, for fixed values $C = \hat{C}$ and $p = \hat{p}$

Then, from (5) and (6), we have

Law 1: Energy-probability Law: For any fixed technology generation (which determines the capacitance $C = \hat{C}$) and constant noise magnitude $\sigma = \hat{\sigma}$, the switching energy $\hat{E}_{\hat{C}, \hat{\sigma}}$ consumed by a probabilistic switch grows with p. Furthermore, the order of growth of $\hat{E}_{\hat{C}, \hat{\sigma}}$ in p is asymptotically bounded below by an exponential in p since $\hat{E}_{\hat{C}, \hat{\sigma}}(p) = \Omega_r\left(E^e_{\hat{C}, \hat{\sigma}}(p)\right)$.

Law 2: Energy-noise Law: For any fixed probability $p = \hat{p}$ and a fixed technology generation (which determines the capacitance $C = \hat{C}$), $\tilde{E}_{\hat{C}, \hat{p}}$ grows quadratically with σ, since $\tilde{E}_{\hat{C}, \hat{p}}(\sigma) = \Omega_r\left(E^q_{\hat{C}, \hat{p}}(\sigma)\right)$.

Earlier forms of these laws [9], [8], [11] were implicitly based on the asymptotic notions described here explicitly for the first time.

IV. IMPLEMENTING APPLICATIONS USING PCMOS TECHNOLOGY

So far, we have summarized abstract models of probabilistic switches and their implementation and characterization in the domain of CMOS. To harness PCMOS technology to implement applications, we will now consider three categories: (i) applications which benefit from (or harness) probabilistic behavior at the device level, (ii) applications that can tolerate probabilistic behavior at the device level and (iii) applications which cannot tolerate statistical behavior. In this paper, we outline our implementation approach for first two categories. For the last category, we envision an approach based on redundancy as well as error correction and recovery techniques, which will be the basis for our future work.

A. Applications that harness probabilistic behavior

We have investigated applications from the cognitive and embedded domain which embody probabilistic behaviors. Probabilistic algorithms are those in which each step, upon repeated execution *with the same inputs*, could have several possible outcomes, where each outcome is associated with a probability parameter. Examples of such algorithms include the celebrated probabilistic tests for primality [16], [17].

Input	Output with corresponding probability parameters		
000	00 (0.98)	01 (0.01)	10 (0.01)
001	00 (0.01)	01 (0.98)	10 (0.01)
010	00 (0.01)	01 (0.01)	10 (0.98)
011	00 (0.98)	01 (0.01)	10 (0.01)
100	00 (0.98)	01 (0.01)	10 (0.01)
101	00 (0.69)	01 (0.30)	10 (0.01)

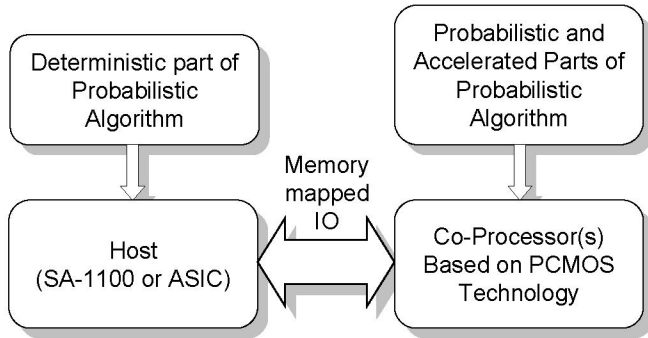Fig. 3. The probabilistic truth table for a node in a Bayesian network with 37 nodes

3

Fig. 4. The canonical PSOC architecture

Algorithm	$\Gamma_{\mathcal{I}}$	
	Min	Max
BN	3	7.43
RNN	226.5	300
PCA	61	82
HE	1.12	1.12

TABLE I

MAXIMUM AND MINIMUM EPP GAINS OF PCMOS OVER THE BASELINE
IMPLEMENTATION WHERE THE IMPLEMENTATION $\mathcal{I}$ HAS A STRONGARM
SA-1100 HOST AND A PCMOS BASED CO-PROCESSOR

In particular, the applications we have considered are based on *Bayesian inference* [18], *Probabilistic Cellular Automata* [19], *Random Neural Networks* [20] and *Hyper Encryption* [21]. For brevity, these algorithms will be referred to as BN, PCA, RNN and HE respectively. Common to these applications (and to almost all probabilistic algorithms) is the notion of a *core probabilistic step* with its associated probability parameter. An abstract model of a core probabilistic step is a probabilistic truth table. In Figure 3, we illustrate the probabilistic truth table for one such step in Bayesian inference. Intuitively, realizing such probabilistic truth tables using probabilistic switches is inherently more efficient with PCMOS switches than with conventional (deterministic) CMOS switches. This is because of the *inherent* probabilistic behavior of the PCMOS switches.

We have implemented these applications as *probabilistic system on a chip* (PSOC) architectures. As illustrated in Figure 4, probabilistic system on a chip architectures are envisioned to consist of two parts: A *host* processor which consists of a conventional low energy embedded processor like the StrongARM SA-1100 [22], coupled to a co-processor which utilizes PCMOS technology and executes the core probabilistic steps.

The *energy-performance product* or EPP is the chief metric of interest for PSOC based architectures [7]; it is the product of the energy and time consumed by an application as it executes on the architecture. Then, for any given application, *energy-performance product gain* $\Gamma_{\mathcal{I}}$ of a PSOC over a conventional (baseline) architecture is the ratio of the EPP of the baseline denoted by β, to the EPP of a particular architectural implementation $\mathcal{I}$. $\Gamma_{\mathcal{I}}$ is thus:

$$\Gamma_{\mathcal{I}} = \frac{Energy_{\beta} \times Time_{\beta}}{Energy_{\mathcal{I}} \times Time_{\mathcal{I}}} \quad (7)$$

When compared to a baseline implementation using software executing on a StrongARM SA-1100, the gain of a PCMOS based PSOC is summarized in Table I

In addition, when the baseline is a custom ASIC realization (host) coupled to a functionally identical CMOS based co-processor, in the context of the HE and PCA applications, the gain $\Gamma_{\mathcal{I}}$ is 9.38 and 561 respectively. Thus, for applications which can harness probabilistic behavior, PSOC architectures

based on PCMOS technology yield several orders of magnitude improvements over conventional (deterministic) CMOS based implementations. For a detailed explanation of the architectures, experimental methodology and a description of the applications, the reader is referred to Chakrapani et. al. [7].

B. Applications that tolerate probabilistic behavior

Moving away from applications that embody probabilistic behaviors naturally (and in turn harness probabilistic behavior of PCMOS devices), we will now consider the domain of applications that *tolerate* probabilistic behavior. Specifically, we investigate applications which can trade energy and performance for application-level quality of the solution. Applications in the domain of digital signal processing are good candidates, where application-level quality of solution is naturally expressed in the form of *signal-to-noise ratio* or SNR. To demonstrate the utility of PCMOS technology in this context, we have implemented filter primitives using PCMOS technology, used to realize the H.264 decoding algorithm [23].

As illustrated in Figure 5(b) the probability parameter p of correctness can be lowered uniformly for each bit in the adder (which is one of the building blocks of the FIR filter used in the H.264 application). While this approach saves energy, it significantly degrades the output picture quality when compared to conventional (CMOS based and error-free) operation. However, as illustrated in Figure 5(c), if the probability parameter is varied *non-uniformly*, significantly lower energy consumption is possible with minimal degradation of the quality of the image [24]. Hence, not only can PCMOS technology be leveraged for implementing energy efficient filters, but also can be utilized to naturally trade-off energy consumed for application level quality of solution through novel probabilistic voltage scaling schemes [24].

V. IMPLEMENTATION CHALLENGES

The actual implementation and fabrication of architectures that leverage PCMOS based devices poses further challenges. Chief among them is "tuning" the PCMOS devices, or in other words, controlling the probability parameter p of correctness. Additionally, the number of distinct probability parameters is a concern, since this number directly relates to the number of voltage levels [9]. We make two observations aimed at addressing these problems: (i) The distinct probability parameters are a requirement of the application and the application *sensitivity*
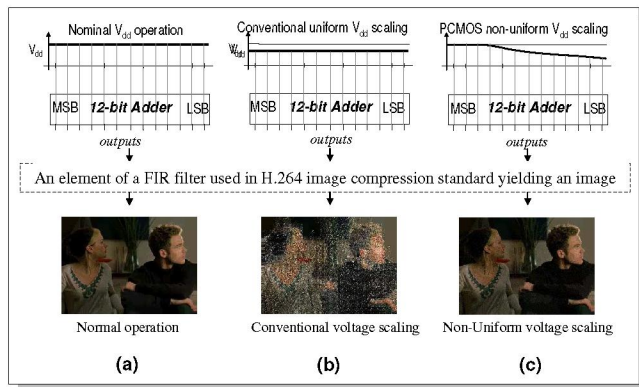
4

Fig. 5. Comparing images reconstructed using H.264 (a) conventional error free operation (b) probability parameter p varied non-uniformly based on bit significance (c)probability parameter p lowered uniformly for all bits

to probability parameters is an important aspect. That is, if an application uses probability parameters p_1, p_2, p_3, for example, it might be the case that the application level quality is not affected when only two distinct values, say p_1, p_2 are used. This, however can only be determined experimentally and is a topic being investigated. (ii) Given probability parameters p_1 and p_2, other probability parameters might be derived through logical operations. For example, if the probability of obtaining a 1 from a given PCMOS device is p and the probability of obtaining a 1 from a second PCMOS device is q, a logical AND of the output of the two PCMOS devices produces a 1 with a probability $p.q$. Using this technique, in the context of an application (the case of Bayesian inference is used here), the number of distinct probability parameters may be drastically reduced. Since the probability parameter p is controlled through varying the voltage, reducing the number of probability parameters reduces the number of distinct voltage levels required and is another topic being investigated.

VI. CONCLUSION AND FUTURE DIRECTIONS

We have demonstrated ways through which probabilistic (noise based) devices can be modeled with considerations for probabilistic behavior as well as energy consumption. We have also demonstrated how these (abstract) models have a physical counterpart in the domain of CMOS and ways through which they can be used to implement applications.

In any implementation of applications which embodies probability, the *quality* of the implementation is an important aspect apart from the energy and running time. In conventional implementations of probabilistic algorithms—which usually leverage hardware or software based implementations of *pseudo* random number generators to supply (pseudo) random bits,—it is a well known fact that random bits of "low quality" affect application behavior, from the correctness of Monte Carlo simulations [25] to the strength of encryption schemes. To ensure that application behavior is not affected by low quality random bits, the quality of random bits produced by a particular strategy should be evaluated rigorously. Our approach to determine the quality of random bits, is to use

statistical tests to determine the quality of randomness. To study the statistical properties of PCMOS devices in a preliminary way, we have utilized the randomness tests from the NIST Suite [26] to assess the quality of random bits generated by PCMOS devices. Preliminary results indicate that PCMOS affords a higher quality of randomness; a future direction of study is to quantify the impact of this quality on the application level quality of solution.

REFERENCES

[1] K. Natori and N. Sano, "Scaling limit of digital circuits due to thermal noise," *Journal of Applied Physics*, vol. 83, pp. 5019–5024, 1998.

[2] N. Sano, "Increasing importance of electronic thermal noise in sub-0.1mm Si-MOSFETs," *The IEICE Transactions on Electronics*, vol. E83-C, pp. 1203–1211, 2000.

[3] L. B. Kish, "End of Moore's law: thermal (noise) death of integration in micro and nano electronics," *Physics Letters A*, vol. 305, pp. 144–149, 2002.

[4] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Proceedings of the 40th Design Automation Conference*, 2003, pp. 338–342.

[5] K. V. Palem, "Proof as experiment: Probabilistic algorithms from a thermodynamic perspective," in *Proceedings of The International Symposium on Verification (Theory and Practice)*, Taormina, Sicily, June 2003.

[6] ——, "Energy aware computing through probabilistic switching: A study of limits," *IEEE Transactions on Computers*, vol. 54, no. 9, pp. 1123–1137, 2005.

[7] L. N. Chakrapani, B. E. S. Akgul, S. Cheemalavagu, P. Korkmaz, K. V. Palem, and B. Seshasayee, "Ultra efficient embedded soc architectures based on probabilistic cmos technology," in *Proceedings of The 9th Design Automation and Test in Europe (DATE)*, Mar. 2006, pp. 1110–1115.

[8] S. Cheemalavagu, P. Korkmaz, and K. V. Palem, "Ultra low-energy computing via probabilistic algorithms and devices: CMOS device primitives and the energy-probability relationship," in *Proceedings of The 2004 International Conference on Solid State Devices and Materials*, Tokyo, Japan, Sept. 2004, pp. 402–403.

[9] S. Cheemalavagu, P. Korkmaz, K. V. Palem, B. E. S. Akgul, and L. N. Chakrapani, "A probabilistic CMOS switch and its realization by exploiting noise," in *Proceedings of The IFIP International Conference on Very Large Scale Integration*, 2005.

[10] J. D. Meindl and J. A. Davis, "The fundamental limit on binary switching energy for terascale integration (TSI)," *IEEE; Journal of Solid State Circuits*, vol. 35, pp. 1515–1516, Oct. 2000.

[11] P. Korkmaz, B. E. S. Akgul, L. N. Chakrapani, and K. V. Palem, "Advocating noise as an agent for ultra low-energy computing: Probabilistic CMOS devices and their characteristics," *Japanese Journal of Applied Physics (JJAP)*, vol. 45, no. 4B, pp. 3307–3316, Apr. 2006.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition*. MIT Press and McGraw-Hill, 2001.

[13] M. O. Rabin, "Complexity of computations," *Communications of the ACM*, vol. 20, no. 9, pp. 625–633, 1977.

[14] J. Hartmanis and R. E. Stearns, "On the computational complexity of algorithms," *Transactions of the American Mathematical Society*, vol. 117, 1965.

[15] N. Ermolova and S. Haggman, "Simplified bounds for the complementary error function; application to the performance evaluation of signal-processing systems," in *Proceedings of the 12th European Signal Processing Conference*, Sept. 2004, pp. 1087–1090.

[16] M. O. Rabin, "Probabilistic algorithms," in *Algorithms and Complexity, New Directions and Recent Trends*, J. F. Traub, Ed., 1976, pp. 29–39.

5

[17] R. Solovay and V. Strassen, "A fast monte-carlo test for primality," *SIAM Journal on Computing*, 1977.

[18] D. MacKay, "Bayesian interpolation," *Neural Computation*, vol. 4, no. 3, 1992.

[19] H. Fuks, "Non-deterministic density classifiation with diffusive probabilistic cellular automata," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 66, 2002.

[20] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution," *Neural Computation*, vol. 1, no. 4, pp. 502–511, 1989.

[21] Y. Z. Ding and M. O. Rabin, "Hyper-Encryption and everlasting security," in *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science; Lecture Notes In Computer Science*, vol. 2285, 2002, pp. 1–26.

[22] Intel Corporation, "SA-1100 microprocessor technical reference manual," Sept. 1998.

[23] D. Marpe, T. Wiegand, and G. J. Sullivan, "The H.264/MPEG4-AVC standard and its fidelity range extensions," *IEEE Communications Magazine*, Sept. 2005.

[24] K. V. Palem, B. E. S. Akgul, and J. George, "Variable scaling for computing elements," *Invention Disclosure*, Feb. 2006.

[25] A. M. Ferrenberg, D. P. Landau, and Y. J. Wong, "Monte carlo simulations: Hidden errors from "good" random number generators," *Phys. Rev. Let*, vol. 69, pp. 3382–3384, 1992.

[26] Random Number Generation and Testing, "http://csrc.nist.gov/rng/."

6