Probabilistic Design: A Survey of *Probabilistic* CMOS Technology and Future Directions for Terascale IC Design

Lakshmi N. B. Chakrapani, Jason George, Bo Marr, Bilge E. S. Akgul, 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 caused by process variations, and other perturbations such as noise. Current circuit design methodologies, which depend on the existence of "deterministic" devices that behave consistently in temporal and spatial contexts do not admit considerations for probabilistic behavior. Admittedly, power or energy consumption as well as the associated heat dissipation are proving to be impediments to the continued scaling (down) of device sizes. To help overcome these challenges, we have characterized CMOS devices with probabilistic behavior (probabilistic CMOS or PCMOS devices) at several levels: from foundational principles to analytical modeling, simulation, fabrication, measurement as well as exploration of innovative approaches towards harnessing them through system-on-a-chip architectures. We have shown that such architectures can implement a wide range of probabilistic and cognitive applications. All of these architectures yield significant energy savings by trading probability with which the device operates correctly—lower the probability of correctness, the greater the energy savings. In addition to these PCMOS based innovations, we will also survey *probabilistic arithmetic*—a novel framework through which traditional computing units such as adders and multipliers can be deliberately designed to be erroneous, while being characterized by a well-defined *probability* of correctness. We demonstrate that in return for erroneous behavior, significant energy and performance gains can be realized through probabilistic arithmetic (units)—over a factor of $4.62X$ in the context of an FIR filter used in a H.264 video decoding—where the gains are quantified through the *energy-performance* product (or EPP). These gains are achieved through a systematic *probabilistic design* methodology enabled by a design space spanning the probability of correctness of the arithmetic units, and their associated energy savings.

1 Introduction and Overview

Device scaling, the primary driver of semiconductor technology advances, faces several hurdles. Manufacturing difficulties in the nanometer regime yield non uniform devices due to parameter variations, and low voltage operation makes them susceptible to

perturbations such as noise [18, 25, 32]. In such a scenario, current day circuit design methodologies are inadequate to design circuits, since they depend on devices with deterministic (in terms of their temporal behavior, since they are operated at high voltages) and uniform spatial behavior. To design robust circuits and architectures in the presence of this (inevitable) emerging statistical phenomena 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 [2].

We have addressed the issue of probabilistic design at several levels: from foundational models [26, 27] of probabilistic *switches* establishing the relationship between probabilistic computing and energy, to analytical, simulation and actual measurement of CMOS devices whose behavior is rendered probabilistic due to noise (which we term as *probabilistic* CMOS, or PCMOS devices). In addition, we have demonstrated design methodologies and practical system-on-a-chip architectures which yield significant energy savings, through judicious use of PCMOS technology, for applications from the cognitive, digital signal processing and embedded domains [3, 13]. In this paper we present a broad overview of our contributions in the area of *probabilistic* design and PCMOS, by surveying prior publications [3, 5, 6, 13, 26, 27]. The exception is our recent work on *probabilistic* arithmetic, a novel framework through which traditional computing units such as adders and multipliers while erroneous, can be used to implement applications from the digital signal processing domain. Specifically, our approach involves creating a novel style of "error-prone" devices with probabilistic characterizations—we note in passing that from a digital design and computing standpoint, the parameter of interest in a PCMOS device is its probability of correctness p—derived by scaling the voltages to extremely and potentially undesirably low levels [19], referred to as *overscaling*.

The rest of the paper is organized as follows. In Section 2 we outline the foundational principles of PCMOS technology based on the *probabilistic Switch*. In Section 3 we show approaches through which 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. The two laws of PCMOS technology using novel asymptotic notions will be the highlights. To help with our exposition, it will be 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 for application categories (i) and (ii), in Section 4.1 and Section 4.2 respectively. In Section 5, we describe probabilistic arithmetic. In Section 6, we outline other emerging challenges such as design for manufacturability, and present a novel probabilistic approach towards addressing one such problem—the problem of multiple voltage levels on a chip. Finally, in Section 7, we conclude and sketch future directions of inquiry.

2 Foundational Principles

Probabilistic switches, introduced by Palem [27], incorporate probabilistic behavior as well as energy consumption as first class citizens and are the basis for PCMOS devices. 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.

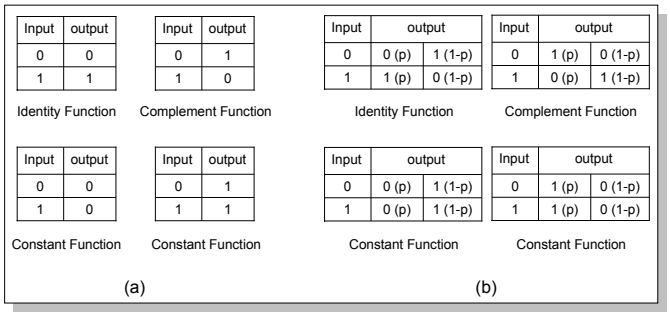

Fig. 1. (a) Deterministic one bit switching functions (b) Their probabilistic counterparts with probability parameter (probability of correctness) p

The relationship between probabilistic behavior—the probability with which the switching steps are correct—and the associated energy consumed was shown to be an entirely novel basis for energy savings [26]. Specifically, principles of statistical thermodynamics were applied to such switches to quantify their energy consumption, and hence the energy consumption (or energy complexity) of a network of such switches. While a switch that realizes the deterministic non-trivial switching function consumes at least $\kappa t \ln 2$ Joules of energy [24], a probabilistic switch can realize a probabilistic non-trivial switching function with $\kappa t \ln(2p)$ Joules of energy in an idealized setting. For a complete definition of a probabilistic switch, the operation of a network of probabilistic switches and a discussion of the energy complexity of such networks, the reader is referred to Palem [27].

3 The CMOS Domain: Probabilistic CMOS

Probabilistic switches serve as a foundational model supporting the physical realizations of highly scaled probabilistic devices as well as emerging devices. In the domain

of CMOS, probabilistic switches model noise-susceptible CMOS (or PCMOS) devices operating at very low voltages [6]. To show that PCMOS based realizations correspond to abstract probabilistic switches, we have identified two key characteristics of PCMOS: (i) probabilistic behavior while switching and (ii) energy savings through probabilistic switching. These characteristics were established through analytical modeling and HSpice based simulations [6, 19] as well as actual measurements of fabricated PCMOS based devices.

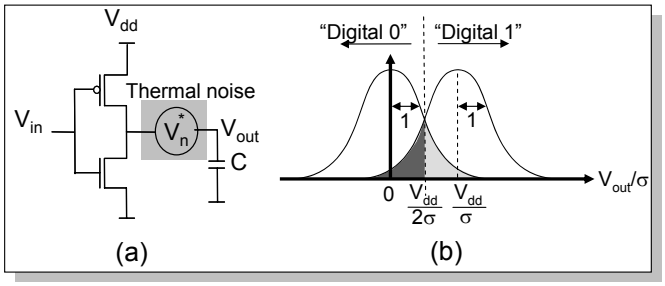

Fig. 2. (a) PCMOS switch (b) Representation of digital values 0 and 1 and the probability of error for a PCMOS switch

For a PCMOS inverter as shown in Figure 2 (a), the output voltage (V_{out}) is probabilistic, in this example, due to (thermal) noise coupled to its output. The associated noise magnitude is statistically characterized by a mean value of 0 and a variance of σ^2. The normalized output voltage $\frac{V_{out}}{\sigma}$ can be represented by a random variable whose value is characterized a Gaussian distribution as shown in Figure 2 (b), where the variance of the distribution is 1. The mean value of the distribution is 0 if the (correct) output is meant to be a digital 0, and $\frac{V_{dd}}{\sigma}$ if the (correct) output is meant 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 associated with this PCMOS inverter during each of its switching steps. From this formulation, we determine the probability of correctness denoted as p, by computing the area in the shaded regions and express p as

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

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

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

Using the bounds for $erfc$ derived by Ermolova and Haggman [9], we have

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

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

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

Clearly, the energy consumed 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 $p = \hat{p}$, $\tilde{E}_{\hat{C},\hat{p}}(\sigma) = \hat{C}\sigma^2 \left(\frac{4}{1.275}\right) \ln\left(\frac{0.28}{1-\hat{p}}\right)$. Similarly for fixed values of $C = \hat{C}$ and $\sigma = \hat{\sigma}$, $\hat{E}_{\hat{C},\hat{\sigma}}$ a function of p alone: $\hat{E}_{\hat{C},\hat{\sigma}}(p) = \hat{C}\hat{\sigma}^2 \left(\frac{4}{1.275}\right) \ln\left(\frac{0.28}{1-p}\right)$.

We will succinctly characterize these behavioral and energy characteristics of PC-MOS switches using asymptotic notions from computer science [7, 14, 30] in the form of two laws. The notion of *asymptotic complexity* is widely used to study the efficiency of algorithms, where "efficiency" is characterized by the growth of its running time (or space), as a function of the size of its inputs [7, 14, 30]. The O notation provides an asymptotic *upper-bound*, where, for a function $f(x)$ where x is an element of the set of natural numbers

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

given any function $h(x)$, there exist positive constants c, x_0 such that $\forall x \geq x_0, 0 \leq f(x) \leq c.h(x)$.

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

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

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

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

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 notation captures 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 $0 \leq g(y) \leq \hat{h}(y)$. Thus our current notion can be interpreted to mean "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.

Returning to the lower-bound from (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 with 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,

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

Observation 1: For $p \in (0, 1)$, whereas the function $E^e_{\hat{C},\hat{\sigma}}(p)$ grows at least exponentially in p, for a fixed $C = \hat{C}$ and $\sigma = \hat{\sigma}$, the function $E^q_{\hat{C},\hat{p}}(\sigma)$, grows at least 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 determined by 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 variations of these laws [5, 6, 19] were implicitly based on the asymptotic notions described here explicitly. Together these laws constitute the characterization of probability and its relationship with energy savings in CMOS devices level. We will now show how this characterization helps build architectures composed of such devices and how energy savings as well as the associated performance gains can be extended up to the application level.

4 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 now reiterate that we consider three application categories: (i) applications which benefit from (or embody) probabilistic behavior intrinsically, (ii) applications that can tolerate probabilistic and (iii) applications which cannot tolerate statistical behavior.

4.1 Applications Which Harness Probabilistic Behavior

We will first consider applications from the cognitive and embedded domains which embody probabilistic behaviors. Probabilistic algorithms are those in which computational steps, upon repeated execution *with the same inputs*, could have distinct outcomes characterized by a probability distribution. A well known example of such an algorithm is the celebrated probabilistic test for primality [29, 34].

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, where the desired probability parameter p is represented parenthetically

In particular, the applications we have considered are based on *Bayesian inference* [21], *Probabilistic Cellular Automata* [11], *Random Neural Networks* [12] and *Hyper Encryption* [8]. 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 such a step is a *probabilistic truth* table. In Figure 3, we illustrate the probabilistic truth table for a step in BN. Intuitively, realizing such probabilistic truth tables using probabilistic switches built from PCMOS is inherently more efficient in terms of the energy consumed when compared to those built from CMOS technology. This is because of the *inherent* probabilistic behavior of the PCMOS switches.

We have constructed *probabilistic system on a chip* (PSOC) architectures for these applications, and 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 [16], coupled to a co-processor which utilizes PCMOS technology and executes the core probabilistic steps.

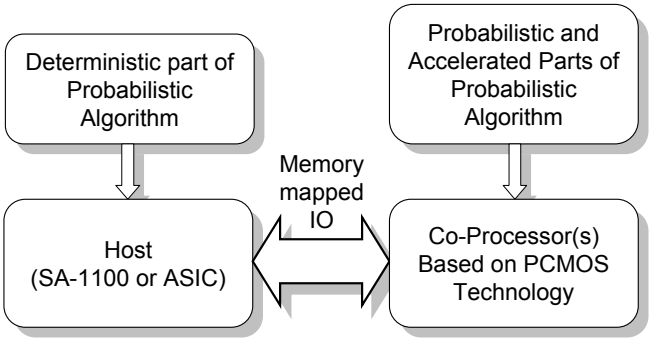

Fig. 4. A canonical PSOC architecture

The *energy-performance product* or EPP is the chief metric of interest for evaluating the efficiency of PSOC based architectures [3]; it is the product of the energy consumed, and time spent in completing an application, as it executes on the architecture. Then, for any given application, *energy-performance product gain* $\Gamma_{\mathcal{I}}$ of its PSOC realization over a conventional (baseline) architecture is the ratio of the EPP of the baseline denoted by the symbol β, to the EPP of a particular architectural implementation $\mathcal{I}$. We note in passing that in the context of the baseline implementation, the source of randomness is a pseudo-random number generator. $\Gamma_{\mathcal{I}}$ is thus:

$$\Gamma_{\mathcal{I}} = \frac{Energy_{\beta} \times Time_{\beta}}{Energy_{\mathcal{I}} \times Time_{\mathcal{I}}} \tag{7}$$

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 1

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

Table 1. Minimum and Maximum 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 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 $\Gamma_{\mathcal{I}}$ improves dramatically to 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. [3].

4.2 Applications Which Tolerate Probabilistic Behavior

Moving away from applications that embody probabilistic behaviors naturally, we will now consider the domain of applications that *tolerate* probabilistic behavior and its associated error. Specifically, we considered applications wherein energy and performance can be traded 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 value of PCMOS technology in one instance, we have implemented filter primitives using PCMOS technology [13], used to realize the H.264 decoding algorithm [23].

As illustrated in Figure 5(b), the probability parameter p_δ 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, the corresponding output picture quality is significantly degraded 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* following the biased method described earlier [28], significantly lower energy consumption can be achieved with minimal degradation of the quality of the image [28]. Hence, not only can PCMOS technology be leveraged for implementing energy efficient filters, but can also be utilized to naturally trade-off energy consumed for application level quality of solution, through novel probabilistic biased voltage scaling schemes [13, 28].

5 Probabilistic Arithmetic

Following our development of characterizing error in the context of probabilistic behaviors induced by noise and considering an adder as a canonical example,we will associate a parameter δ, which indicates the magnitude by which the output of a computing element, an adder for example, can deviate from the correct answer before it is deemed to be erroneous; thus, an output value that is within a magnitude of δ from the correct value is declared to be correct. The rationale for this approach is that in several embedded domains in general and the DSP domain in particular—a topic to be discussed in some detail in the sequel—error magnitudes below a "tolerable"' threshold, quantified through δ, will be shown to have no impact on the perceived quality of an image. The *probability* of correctness p_δ of the *probabilistic adder* is defined, following the frequentist notion, to be the ratio of the number of correct values of the output compared to the total number of values.

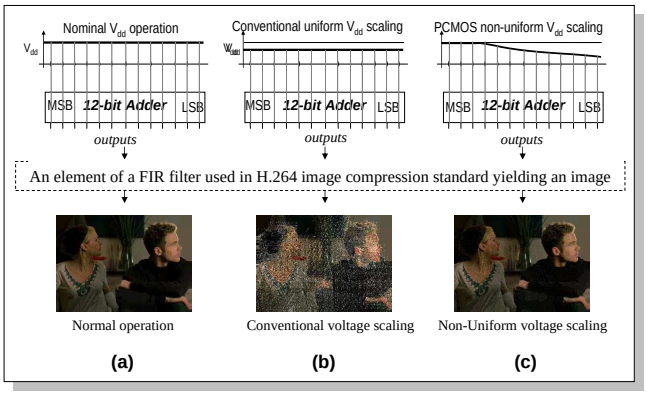

Fig. 5. Comparing images reconstructed using H.264 decoder (a) Conventional error-free operation (b) Probability parameter p_δ lowered uniformly for all bits (c) Probability parameter p_δ varied non-uniformly based on bit significance

Note that our approach to realizing energy savings and performance gains is entirely novel, and can be distinguished from similar aggressive and well-known approaches to voltage scaling: Our approach is aimed at designing arithmetic elements that are *deliberately designed to function in an erroneous manner*, albeit in a regime where such erroneous behavior can be characterized through probabilistic models and methods.

5.1 Probabilistic Arithmetic Through Voltage Overscaling

It is well known that energy savings can be achieved through scaling down supply voltages in a circuit. However, in the past, this approach resulted in increasing propagation times, consequently lowering the circuit's performance. Instead of avoiding bit errors through conventional voltage scaling, we advocate a probabilistic approach through voltage overscaling. We consider an approach here wherein the speed of the system clock is not lowered, even as the switching speed of the data path is lowered by voltage scaling.

Thus, source of error is caused by the gap between the values of the clock period, γ, and the effective switching speed σ. To understand this point better, consider keeping the system clock period, γ, fixed at $6ns$ in the context of a "probabilistic adder"; throughout this paper, we will consider the ripple-carry algorithm [20] for digital addition. Now, consider that as a result of voltage overscaling, for a particular input, the adder switches at a slower clock value of $\sigma = 7ns$, potentially yielding an incorrect result when its output is consumed or read by the system every $6ns$. Thus the output is not completely calculated at $6ns$ (system clock speed) intervals since the adder takes $7ns$ to completely switch in this example. However, by lowering the operating voltage of the adder and thus increasing σ, the energy consumption of the adder is lowered as well. In this case, the relationship of interest is between the rate at which the output value is incorrect and the associated savings in energy. As one would expect, error rates

will be increased while yielding greater energy savings and this relationship will be characterized in Section 5.2.

5.2 Energy Savings Through Overscaled PCMOS

Typically, the nominal clock rate for a computing element is set by allowing for the worst case, critical path delay. However, the critical path is not active for most operational data sets, since the active path in the circuit is determined by the input data. In order to maintain correct operation, all potential propagation paths must be considered and the system clock rate must accommodate this worst case. This results in a clock period to delay (*clock-to-delay*) gap necessary to account for worst case. Voltage overscaling, however, attempts to take advantage of this gap by trading deterministic operation in exchange for energy savings.

Empirically Characterizing the Energy-Probability Relationship Through Benchmarks To demonstrate the potential energy savings through overscaled PCMOS, we consider an 18-bit ripple-carry adder, a 9-bit two's-complement tri-section, array multiplier, and a 6-tap 9-bit FIR filter composed of adders and multipliers. In each of these cases, we will execute three benchmark data sets: (i) uniformly distributed random data, (ii) H.264 data from a low quality video source, and (iii) H.264 data from a high quality video source. As seen in Table 2, all three cases show reductions in energy consumption. However, H.264 data sets yield greater energy reductions when compared to uniformly distributed data. This is due to the fact that H.264 video data tends to have little variance and relatively infrequent output switching and as a result, only small portions of the circuit are active on occasions when there is output switching. As a result, the computation infrequently causes delays greater than the system clock period.

Conversely, uniformly distributed data exercises all portions of the circuit because of an associated larger variance. Accordingly, there is a smaller clock-to-delay gap and as a result, the energy savings are lower for a given probability parameter p.

Table 2. Voltage Overscaled PCMOS Energy Savings for Benchmark Data Sets

Computing Element	Benchmark	p_δ	$E(p_\delta)$	Δp_δ	$\Delta E(p_\delta)$	p_δ Sacrifice	Energy Savings
Adder	Uniform data	0.9999	$0.88pJ$	0.0001	$2.59pJ$	0.01%	75%
$E_{nom} = 3.47pJ$	Low Quality H.264	0.9993	$0.62pJ$	0.0007	$2.85pJ$	0.07%	82%
δ threshold $= 127$	High Quality H.264	0.9998	$0.62pJ$	0.0002	$2.85pJ$	0.02%	82%
Multiplier	Uniform data	0.9998	$8.30pJ$	0.0002	$11.73pJ$	0.02%	59%
$E_{nom} = 20.03pJ$	Low Quality H.264	0.9549	$2.11pJ$	0.0451	$17.92pJ$	4.51%	89%
δ threshold $= 127$	High Quality H.264	0.9862	$2.11pJ$	0.0138	$17.92pJ$	1.38%	89%
FIR	Uniform data	0.9999	$102.89pJ$	0.0001	$34.67pJ$	0.01%	25%
$E_{nom} = 137.56pJ$	Low Quality H.264	0.9998	$37.37pJ$	0.0002	$100.19pJ$	0.02%	73%
δ threshold $= 255$	High Quality H.264	0.9999	$57.46pJ$	0.0001	$80.1pJ$	0.01%	58%

As in the case of PCMOS devices, the energy-probability relationship will be used to characterize our design space. As an illustrative example, we will consider an 18 bit ripple carry adder and its overscaled variants. The design space is characterized by three dimensions. The probability parameter p_δ, the energy and the relationship between γ and σ. For example in Figure 6, consider a specific value of energy. For this fixed energy budget, the probability of correctness is determined by the clock period of the circuit. As a result of these three properties, there exists a 3-dimensional design space where probability of correct output can be traded for energy savings and performance gains. A plot of one possible design space for a PCMOS adder is shown in Figure 6.

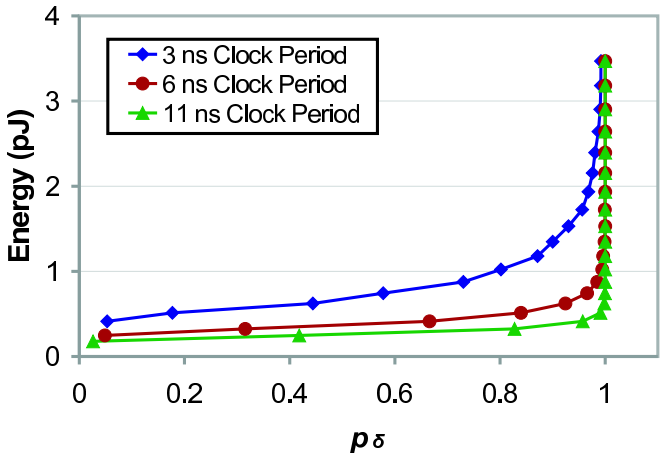

Fig. 6. Energy/performance/probability tradeoff for an 18-bit, ripple-carry adder: at nominal clock rate (11 ns period), at $1.8X$ faster clock rate (6 ns period), and at $3.7X$ faster clock rate (3 ns period)

By extension, energy can be saved and performance improved by increasing the error rate p_δ. This novel approach to achieving significant energy savings are possible since a "small" decrease in the probability of correctness can yield a disproportionate gain in energy savings (Table 3) as well as in the associated EPP. This energy-probability tradeoff is also characterized in Section 5.2 through the energy-probability or E-p relationship of elemental gates used to realize probabilistic arithmetic. Through this relationship, we provide a coherent characterization of the design space associated with probabilistic arithmetic. Specifically, the design space is determined by the parameters γ and σ yielding a probability parameter p_δ, with an associated energy consumption $E(p_\delta)$.

Using this notion of probabilistic arithmetic primitives as building blocks, we implement two widely used DSP algorithms: the fast Fourier transform (FFT) and the finite impulse response (FIR) filter. As a result of the probabilistic behavior of the arithmetic primitives, the associated DSP algorithm computations are also probabilistic. In this paper, we show the EPP gains in the context of the FIR filter in Section 5.2, and extend it to

Table 3. Probability of correctness and energy savings for a PCMOS adder

Benchmark	p_δ Degradation	Energy Savings
Low Quality Video	0.07%	82%

demonstrate gains at the application level in the context of a movie decoded using this filter based on the H.264 standard. Briefly, from the perspective of human perception, the degradation in quality is negligible whereas the gains quantified through the EPP metric were a factor of $3.70X$ as presented in Section 5.2).

There are several subtle issues that have played a role in this formulation, notably the ability to declare a phenomenon—the behavior of adder in our case—to be *probabilistic* based on aposteriori statistical validation. A detailed analysis is beyond the scope of this discussion, and the interested reader is referred to Jaynes's excellent treatment of this topic [17].

Case Study of an FIR To analyze the value and the concomitant savings derived from voltage overscaled PCMOS, we have evaluated H.264 video decoding algorithm. Motion compensation is a key component of the H.264 decoding algorithm. Within this motion compensation phase a six-tap FIR is used to determine luminosity for the H.264 image blocks using 1, -5, 20, 20, -5, and 1 as the coefficients at taps $[0..5]$ respectively. Video data from a low quality source (military video of ordnance explosion) and a high quality source (video from the 20th Century Fox movie XMen 2) were used for experimentation.

Experimental Framework First, the FIR was decomposed into its constituent adder and multiplier building blocks. These building blocks were then decomposed into full adders classified by type and output loading. Each full adder class was then simulated in HSpice for all input state transitions that result in an output state transition. This was repeated for both the sum and carry out bits of the full adder classes, and the resulting output transition delays were then summarized into a transition-delay lookup table. All input state transitions that did not result in an output state transition were considered to have no delay. HSpice simulation was then repeated with 1000 uniformly distributed random input combinations for each full adder class to determine average switching energy.

Building on this HSpice model and using a C-based simulation, benchmark data was used and using the current and previous states for both input and output at each full adder, the delay is estimated for each model using the look-up table previously developed using the HSpice simulation framework. Individual full adder delays were further propagated to building block outputs, which were then propagated to FIR outputs and compared to a specified clock period γ. Any FIR output delays violating timing constraints were considered to be erroneous and the appropriate bit was deemed incorrect and forced to be erroneous. The results of the outputs of the FIR filter in the fully functional context is then compared to those derived from overscaling to determine p_δ and

SNR. Energy consumption was determined by adding the energy of each individual full adder comprising the FIR and the results were compared to conventional operation (at a supply voltage V_{dd}=$2.5V$). The overall delay in the FIR filter was determined by maximum propagation delay calculated as the sum of worst case delays for each full adder in the critical path.

Finally, H.264 decoding was performed using a program written by Martin Fiedler. The original code was modified to inject bit-errors determined by the C simulation described above. The resulting decoded frames were then compared to originals to determine SNR. Energy consumption was calculated as the FIR energy consumption for the specific voltage overscaling scheme employed.

FIR **Results** As shown in Figure 7, voltage overscaled PCMOS operation yielded a 47% reduction in energy consumption with a $2X$ factor increase in performance, resulting in an EPP ratio of $3.70X$ for high quality video. We also consider a low quality military video, where the primary requirement is object recognition, and larger gains in energy savings and performance are possible. Thus, voltage overscaled PCMOS operation yeilds a 57% reduction in energy consumption and $2.19X$ factor increase in performance gain with an EPP ratio of $4.62X$ in this case where the quality of the output video is not as significant as the high quality case.

Fig. 7. Application level impact of our approach on high quality H.264 video comparing voltage scaled PCMOS [bottom](with an EPP ratio of $3.70X$) to the original H.264 frames [top]

6 Related work and Some Implementation Challenges

The use of voltage scaling in an effort to reduce energy consumption has been explored vigorously in previous work [4, 22, 36, 37]. In each of these papers, increased propagation delay was considered the primary drawback to voltage overscaling. To maintain circuit performance and *correctness* while simultaneously realizing energy savings through voltage scaling, several researchers employ the use of multiple supply voltages by operating elements along the critical path at nominal voltage and reducing supply

voltages along non-critical paths [4, 22, 36, 37]. Supply voltage scheduling and its interplay with path sensitization along with task scheduling has been studied as well [4, 22, 36].

Offering a contrasting approach, in [15, 33, 35], propagation delay errors are removed through error correction in a collection of techniques named "algorithmic noise-tolerance (ANT)". In [15], difference-based and prediction-based error correction approaches are investigated and in [35], adaptive error cancellation (AEC) is employed using a technique similar to echo cancellation. In [33], the authors propose reduced precision redundancy (RPR) to eliminate propagation delay errors with no degradation to the SNR of the computed output. Our work can be distinguished from all of these methods through the fact that our designs permit the outputs of the arithmetic units to be incorrect, albeit with a well-understood probability.

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 [6]. We make two observations aimed at addressing these problems: (i) Having distinct probability parameters is a requirement of the application and the application *sensitivity* 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, this, in turn reduces the number of distinct voltage levels required and is another topic being investigated.

7 Remarks on Quality of Randomness and Future Directions

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 utilize 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 [10] 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 [31] 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.

Acknowledgments

This work is supported in part by DARPA under seedling contract #F30602-02-2-0124, by the DARPA ACIP program under contract #FA8650-04-C-7126 through a subcontract from USC-ISI and by an award from Intel Corporation. This document is an expansion of the survey originally presented at the IFIP international conference on very large scale integration [1] and includes novel results.

References

1. B. E. S. Akgul, L. N. Chakrapani, P. Korkmaz, and K. V. Palem. Probabilistic CMOS technology: A survey and future directions. In *Proceedings of The IFIP International Conference on Very Large Scale Integration*, 2006.
2. 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*, pages 338–342, 2003.
3. 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)*, pages 1110–1115, Mar. 2006.
4. J. Chang and M. Pedram. Energy minimization using multiple supply voltages. In *Proc. of IEEE Transactions on VLSI Systems*, volume 5, pages 436 – 443, Dec. 1997.
5. 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*, pages 402–403, Tokyo, Japan, Sept. 2004.
6. 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.
7. T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. MIT Press and McGraw-Hill, 2001.
8. 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*, volume 2285, pages 1–26, 2002.
9. 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* 12^{th} *European Signal Processing Conference*, pages 1087–1090, Sept. 2004.
10. A. M. Ferrenberg, D. P. Landau, and Y. J. Wong. Monte carlo simulations: Hidden errors from "good" random number generators. *Phys. Rev. Let*, 69:3382–3384, 1992.
11. H. Fuks. Non-deterministic density classifiation with diffusive probabilistic cellular automata. *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, 66, 2002.
12. E. Gelenbe. Random neural networks with negative and positive signals and product form solution. *Neural Computation*, 1(4):502–511, 1989.

13. J. George, B. Marr, B. E. S. Akgul, and K. Palem. Probabilistic arithmetic and energy efficient embedded signal processing. In *International Conference on Compilers, Architecture, and Synthesis for Embedded Systems CASES*, 2006.
14. J. Hartmanis and R. E. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117, 1965.
15. R. Hedge and N. R. Shanbhag. Soft digital signal processing. *IEEE Transactions on VLSI*, 9(6):813 – 823, Dec. 2001.
16. Intel Corporation. SA-1100 microprocessor technical reference manual, Sept. 1998.
17. E. Jaynes. *Probability Theory: The Logic of Science*. Cambridge University Press, Cambridge, UK, 2003.
18. L. B. Kish. End of Moore's law: thermal (noise) death of integration in micro and nano electronics. *Physics Letters A*, 305:144–149, 2002.
19. 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)*, 45(4B):3307–3316, Apr. 2006.
20. M. Lu. *Arithmetic and Logic in Computer Systems*. John Wiley & Sons, Inc., Hoboken, NJ, 2004.
21. D. MacKay. Bayesian interpolation. *Neural Computation*, 4(3), 1992.
22. A. Manzak and C. Chaktrabarti. Variable voltage task scheduling algorithms for minimizing energy/power. In *Proc. of IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, volume 11, pages 270 – 276, Apr. 2003.
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. J. D. Meindl and J. A. Davis. The fundamental limit on binary switching energy for terascale integration (TSI). *IEEE; Journal of Solid State Circuits*, 35:1515–1516, Oct. 2000.
25. K. Natori and N. Sano. Scaling limit of digital circuits due to thermal noise. *Journal of Applied Physics*, 83:5019–5024, 1998.
26. 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.
27. K. V. Palem. Energy aware computing through probabilistic switching: A study of limits. *IEEE Transactions on Computers*, 54(9):1123–1137, 2005.
28. K. V. Palem, B. E. S. Akgul, and J. George. Variable scaling for computing elements. *Invention Disclosure*, Feb. 2006.
29. M. O. Rabin. Probabilistic algorithms. In J. F. Traub, editor, *Algorithms and Complexity, New Directions and Recent Trends*, pages 29–39. 1976.
30. M. O. Rabin. Complexity of computations. *Communications of the ACM*, 20(9):625–633, 1977.
31. Random Number Generation and Testing. http://csrc.nist.gov/rng/.
32. N. Sano. Increasing importance of electronic thermal noise in sub-0.1mm Si-MOSFETs. *The IEICE Transactions on Electronics*, E83-C:1203–1211, 2000.
33. B. Shim, S. R. Sridhara, and N. R. Shanbhag. Reliable low-power digital signal processing via reduced precision redundancy. In *Proc. of IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, volume 12, pages 497 – 510, May 2004.
34. R. Solovay and V. Strassen. A fast monte-carlo test for primality. *SIAM Journal on Computing*, 1977.
35. L. Wang and N. R. Shanbhag. Low-power filtering via adaptive error-cancellation. *IEEE Transactions on Signal Processing*, 51:575 – 583, Feb. 2003.
36. Y. Yeh and S. Kuo. An optimization-based low-power voltage scaling technique using multiple supply voltages. In *Proc. of IEEE Internaitonal Symposium on ISCAS 2001*, volume 5, pages 535 – 538, May 2001.

37. Y. Yeh, S. Kuo, and J. Jou. Converter-free multiple-voltage scaling techniques for low-power cmos digital design. In *Proc. of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, volume 20, pages 172 – 176, Jan. 2001.