Sustaining Moore's Law in Embedded Computing through Probabilistic and Approximate Design: Retrospects and Prospects

Krishna V. Palem
VISEN Center
Rice University
Houston, Texas, USA
palem@cs.rice.edu

Lakshmi N.B. Chakrapani
VISEN Center
Rice University
Houston, Texas, USA

Zvi M. Kedem
Courant Institute of
Mathematical Sciences
New York University
New York, New York, USA

Avinash Lingamneni
VISEN Center
Rice University
Houston, Texas, USA

Kirthi Krishna Muntimadugu
VISEN Center
Rice University
Houston, Texas, USA

ABSTRACT

The central theme of our work is the probabilistic and approximate design of embedded computing systems. This novel approach consists of two distinguishing aspects: (*i*) the design and implementation of embedded systems, using components which are susceptible to perturbations from various sources and (*ii*) a design methodology which consists of an exploration of a design space which characterizes the trade-off between quality of output and cost, to implement high performance and low energy embedded systems. In contrast with other work, our design methodology does not attempt to correct the errors introduced by components which are susceptible to perturbations, instead we design "good enough" systems. Our work has the potential to address challenges and impediments to Moore's law arising from material properties and manufacturing difficulties, which dictate that we shift from the current-day deterministic design paradigm to statistical and probabilistic designs of the future. In this paper, we provide a broad overview of our work on probabilistic and approximate design, present novel results in approximate arithmetic and its impact on digital signal processing algorithms, and sketch future directions for research.

Categories and Subject Descriptors

C.3 [**Computer Systems Organization**]: Special-Purpose and Application-Based Systems—*Signal processing systems; Real-time and embedded systems*

CASES'09, October 11–16, 2009, Grenoble, France.
Copyright 2009 ACM 978-1-60558-626-7/09/10 ...$10.00.

General Terms

Design, Experimentation, Performance

Keywords

Probabilistic design, Digital signal processing, Approximate design, Probabilistic arithmetic, Approximate arithmetic, Probabilistic CMOS

1. INTRODUCTION

Typically, embedded computing systems are required to achieve a required level of computing performance, with simultaneous and severe constraints on their characteristics such as power consumption, mobility and size. Moore's law and the associated shrinking of transistor sizes, increase in mobility, decrease in size and power consumption has served as a driver for the proliferation and ubiquity of embedded systems. It is desirable for this trend to continue, to enable new applications and novel contexts in which embedded systems could be used. However, our ability to miniaturize silicon-based transistors is under serious jeopardy. These challenges can broadly be classified under two categories (*i*) the change in the nature of materials and material properties as the sizes of the transistors decrease. and (*ii*) our inability to fabricate identical and reliable nanometer-sized silicon devices and achieve uniform behavioral characteristics. These challenges affect the physical characteristics of transistors and hence computing platforms in many ways [3]. For example, devices are no longer expected to behave in a deterministic and reliable manner, and the probabilistic and unreliable behavior of devices is deemed inevitable by the international technology road-map for semiconductors (ITRS) which forecasts [16] *"Relaxing the requirement of 100% correctness for devices and interconnects may dramatically reduce costs of manufacturing, verification, and test. Such a paradigm shift is likely forced in any case by technology scaling, which leads to more transient and permanent failures of signals, logic values, devices, and interconnects."*. This non-uniform, probabilistic and unreliable behavior of transistors has an impact on the desirable characteristics of embedded systems. For example, to provide adequate noise immunity, the supply voltage of transistors are not scaled down at a rate concomitant to the reduction of the size of the transistors [13]. This results in an increase in power density as size of the transistors decrease without a corresponding decrease in power consumption. Increasing power density results in bulky cooling components thus severely impacting the mobility of embedded computing platforms. A comprehensive survey of such challenges to nanometer-sized devices and beyond may be found in [3, 4, 5, 13]. Several approaches have been adopted to address these challenges to Moore's law. These approaches include rigorous test mechanisms, techniques which correct errors incurred by architectural primitives using temporal and spatial redundancy [2, 6, 14], an increase in parallelism without an increase in the frequency of operation of computing devices, research into novel non-silicon materials for computing, including molecular devices [33], graphene and optoelectronics, and design automation-based approaches to reduce the impact of undesirable effects such as timing variations and noise susceptibility.

By contrast, the central theme of our work, which we refer to as *probabilistic and approximate design*, is to *design computing systems using circuit components which are susceptible to perturbations*. We use the term "perturbations" to cover a broad range of phenomena which cause circuit elements to behave in an "incorrect" or "non-uniform" manner. These behaviors may arise from unreliable behavior of computing devices—caused by say, susceptibility due to noise—or unpredictable behavior due to variations in the delay of circuit elements. We note that a salient feature of our work is that we do not attempt to correct errors incurred by circuit elements, instead we use them in the context of applications which can benefit from or tolerate such behaviors. Our research on probabilistic and approximate design has three inter-related aspects drawing on theoretical background from diverse disciplines such as probabilistic algorithms, theory of digital signal processing, thermodynamics, computer arithmetic and mathematical logic. The three aspects are

1. *Applications:* In general, the set of all embedded applications may be classified under three categories. Those which benefit from perturbations, those which can tolerate but do not benefit from perturbations, and those which cannot tolerate perturbations. Our work aims to implement applications derived from the former two categories. This will be expanded upon in Section 2.

2. *Device properties:* In the application context outlined above, energy and performance efficiency can be obtained if the computing devices used for the implementation provide some mechanism through which "correctness" may be traded for cost. In our work, though our principles are general, we consider implementations based on complementary metal oxide semiconductor (CMOS) technology. In this context, we have considered two phenomena in CMOS: (*i*) the relationship between the probability of correct switching and energy consumption. This relationship will be expanded upon in Section 3.1 and (*ii*) the relationship between the speed of switching and the supply voltage of CMOS-based logic gates.

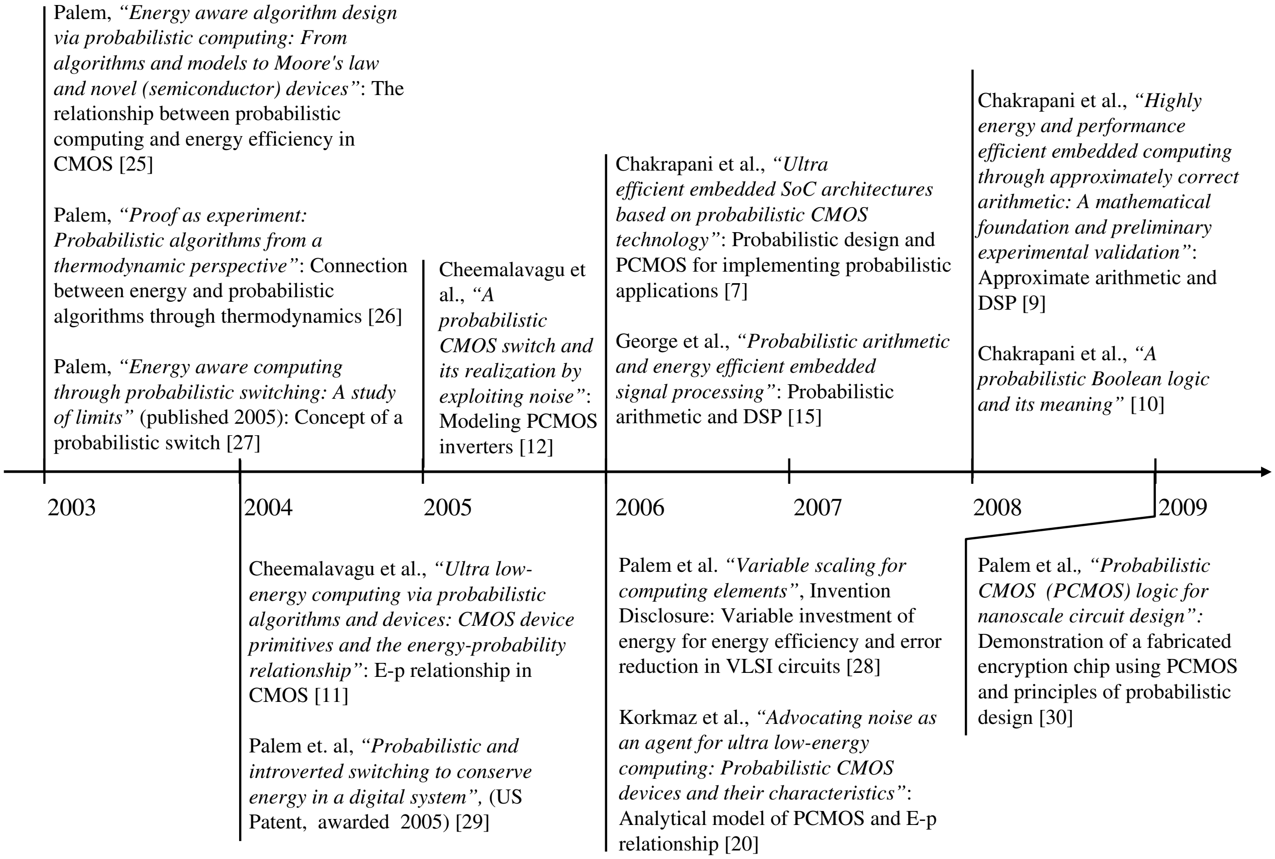
3. *Design practice:* Given applications which can benefit from or can tolerate perturbations, and CMOS devices which exhibit a trade-off between perturbations and cost, we need a design methodology through which these applications may be implemented using these computing devices. Our design methodology is rooted in the theory of *probabilistic Boolean logic* (PBL), *probabilistic arithmetic* and *approximate arithmetic*.

We distinguish between *probabilistic design* where the behavior of the computing substrate is probabilistic, and *approximate design*, where the behavior of the computing substrate is deterministic, but erroneous. The first discussion on *probabilistic design* spans Section 3. In the second thread on *approximate design* spanning Section 4, we survey our work which uses the relationship between energy consumption and switching speed of CMOS devices to achieve a cost-quality trade-off in DSP applications. We note that the former (probabilistic design) approach is relevant for future technology generations where noise is likely to be comparable to signal levels and the latter (approximate design) approach based on conventional CMOS technology is relevant for energy efficient implementations using current-day technology generations.

1.1 A Chronology

A chronology of a representative set of our publications is shown in Figure 1. With the connection between computing and energy as a background [21], in 2003, Palem posited a connection between *probabilistic* computing and energy consumption [25]. In particular, he showed that probabilistic algorithms are *inherently* efficient when compared to their deterministic counterparts [26]. He introduced a model of computing, called the randomized bit-level random access machine (RaBRAM) and showed how probabilistic algorithms may be implemented on this model in an energy efficient manner. As an extension to Landauer's and Meindl's work on deterministic switching, Palem [27] introduced the *probabilistic* switch. A probabilistic switch is a computational device that computes one of the four possible 1-bit input and 1-bit output functions with an associated probability of correctness $\frac{1}{2} \leq p \leq 1$. Through the principles of thermodynamics, Palem showed that $kT \ln(2p)$ joules of energy is sufficient for one switching of a probabilistic switch [27]. Building on this, he introduced the *network of switches* model of computing and showed how logical operations such as disjunction, conjunction and negation, each with an associated probability of correctness, may be computed through the composition of probabilistic switches. This is the first characterization of the use of an inaccurate (or incorrect) device to achieve a trade-off between (energy) investment and correctness. In 2004, again as shown in Figure 1, based on these advances, Cheemalavagu et al. showed that this relationship between energy and probability of correctness could be achieved in the domain of CMOS [11, 12]. The use of these *probabilistic* CMOS devices and a VLSI-based network of switches were identified as a novel technique for energy efficiency and a patent [29] was awarded in 2005.

Moving forward to the year 2006 in Figure 1, Korkmaz et al. developed analytical models to relate the energy consumption of PCMOS inverters to their probability of correctness [20]. The relationship between energy and probability of correct switching is expanded upon in Section 3.1. Based on the RaBRAM model and the network of switches, Chakrapani et al. demonstrated how a *probabilistic System-on-a-chip* architecture may be designed using PCMOS devices, and to implement probabilistic applications (which *benefit* from perturbations) in an efficient manner [7]. An ex-



Palem, *"Energy aware algorithm design via probabilistic computing: From algorithms and models to Moore's law and novel (semiconductor) devices"*: The relationship between probabilistic computing and energy efficiency in CMOS [25]

Palem, *"Proof as experiment: Probabilistic algorithms from a thermodynamic perspective"*: Connection between energy and probabilistic algorithms through thermodynamics [26]

Palem, *"Energy aware computing through probabilistic switching: A study of limits"* (published 2005): Concept of a probabilistic switch [27]

Cheemalavagu et al., *"Ultra low-energy computing via probabilistic algorithms and devices: CMOS device primitives and the energy-probability relationship"*: E-p relationship in CMOS [11]

Palem et. al, *"Probabilistic and introverted switching to conserve energy in a digital system",* (US Patent, awarded 2005) [29]

Cheemalavagu et al., *"A probabilistic CMOS switch and its realization by exploiting noise"*: Modeling PCMOS inverters [12]

Chakrapani et al., *"Ultra efficient embedded SoC architectures based on probabilistic CMOS technology"*: Probabilistic design and PCMOS for implementing probabilistic applications [7]

George et al., *"Probabilistic arithmetic and energy efficient embedded signal processing"*: Probabilistic arithmetic and DSP [15]

Palem et al. *"Variable scaling for computing elements"*, Invention Disclosure: Variable investment of energy for energy efficiency and error reduction in VLSI circuits [28]

Korkmaz et al., *"Advocating noise as an agent for ultra low-energy computing: Probabilistic CMOS devices and their characteristics"*: Analytical model of PCMOS and E-p relationship [20]

Chakrapani et al., *"Highly energy and performance efficient embedded computing through approximately correct arithmetic: A mathematical foundation and preliminary experimental validation"*: Approximate arithmetic and DSP [9]

Chakrapani et al., *"A probabilistic Boolean logic and its meaning"* [10]

Palem et al., *"Probabilistic CMOS (PCMOS) logic for nanoscale circuit design":* Demonstration of a fabricated encryption chip using PCMOS and principles of probabilistic design [30]

Figure 1: A chronology of representative publications on probabilistic and approximate design

panded version of the work may be found in [8]. George et al. demonstrated how probabilistic devices may be used in the context of digital signal processing applications which can *tolerate* perturbations [15]. This approach based on probabilistic arithmetic is expanded upon in Section 3.2.2. A key contribution of this work was the idea of investment of energy in circuit elements, in a way which is proportional to the value of the output that they compute. This was filed [28] as an invention disclosure in 2006.

In 2008, we developed a new logic, the *probabilistic Boolean logic* to study the theoretical properties and meaning of circuits composed of probabilistic elements [10]. An overview of the salient aspects of this logic, which is relevant to probabilistic design is given in Section 3.2.1. Thus in our work surveyed above, two foundational ideas were introduced (*i*) the fact that logical operations with an associated probability of correctness p, can be inherently energy efficient than their deterministic counterparts and (*ii*) Such operations may be used to implement probabilistic algorithms to achieve energy efficiency in a novel way. The principles of probabilistic design were applied to current-day technology generations to design *approximately correct* arithmetic circuits. A description of this work can be found in [9] and an overview is in Section 4. In 2009, the test results of a PCMOS-based encryption chip was presented [30] demonstrating the energy and performance efficiency of PCMOS-based VLSI circuits [1].

2. THE APPLICATION CONTEXT

The use of randomness to achieve efficient computation has been studied in the context of probabilistic algorithms such as the randomized test for primality discovered by Rabin [31], Solovay and Strassen [32]. Probabilistic algorithms, which consume "random bits" or "coin tosses", have been shown to be more efficient than their deterministic counterparts (in terms of the running time and storage space), in solving a wide variety of problems. Such algorithms typically embody non-deterministic behavior in terms of their running time or correctness of their output. Hence in this context, applications which are based on probabilistic algorithms may *benefit* from randomness by harnessing randomness to achieve efficiency.

A second category of applications—applications based on digital signal processing (DSP) is an example—can *tolerate* occasional perturbations introduced due to erroneous computation. This may be due to two reasons (*i*) the limited sensitivity and the psycho-biological tolerance of human senses. This is relevant when these applications are used in the context of a system whose output is perceived by humans. A digital signal processing application which decodes compressed video in a mobile video-player is a good example and (*ii*) The errors introduced by the sensors and the environment might be the dominant factor which determines the quality of the output. For example, in a synthetic aperture RADAR signal processing application which processes noisy RADAR data, the noise introduced by the sensor and

[1] For a complete bibliography, please visit http://visen.rice.edu/papers.aspx

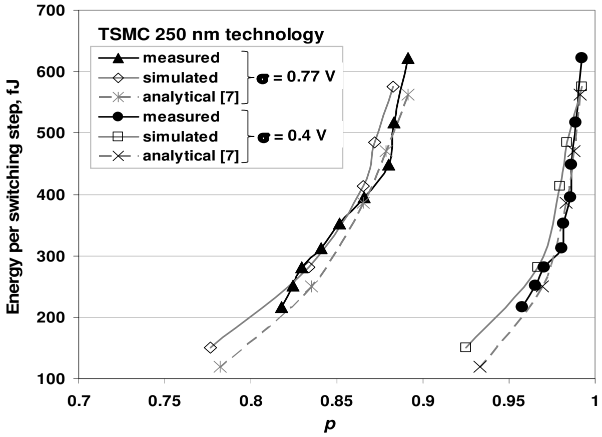


Figure 2: The E-p relationship of PCMOS inverters. Measurement, simulation, and analytical results based on TSMC $0.25\mu m$ technology with a noise magnitude of $0.77V$ RMS and $0.4V$ RMS. (from [30] and [19])

the environment might be the dominant factor which determines the quality of the output. These two categories of applications will be of interest in our work. In the latter class of DSP-based applications, we will use the signal to noise ratio (SNR) as well as the *perceived quality* as a metric to determine the performance of such applications.

3. PROBABILISTIC DESIGN

In the context of applications which can tolerate errors introduced by the computational elements, let us term this relaxed requirement of correctness as the "slack" in the application (we shall formalize this in Section 3.2.1). If this "slack" in the requirement of correctness were be exploited to yield energy and performance benefits, a quality-cost trade-off can be achieved. To achieve this quality-cost trade-off, some property of the technology in which these computational elements are implemented (in our context, it is CMOS-based VLSI circuits) which facilitates such a trade-off needs to be identified and utilized. We consider two such properties, the *Energy-probability of correctness* relationship, which we term as the E-p relationship and the relationship between the speed of switching and the supply voltage (and hence the energy consumption) in conventional CMOS devices.

3.1 The E-p Relationship

The relationship between energy and probability of correct switching, which we call the E-p relationship, extends to the domain of CMOS as well. Analytical modeling and simulation by Cheemalavagu et al. [11] demonstrated that to increase the probability of correct switching, CMOS devices should be operated at higher voltages, thereby incurring higher energy consumption for each switching step. The devices considered in that work were rendered probabilistic due to perturbations from thermal noise (it has been projected that the level of thermal noise would be comparable to signal levels in deeply scaled silicon devices [18, 22]). We refer to such probabilistic devices as *probabilistic* CMOS technology, or PCMOS technology for short. Our technique for implementing such devices is described in [29]. Korkmaz et al. studied this relationship and developed a model to relate the energy consumption to the probability of correctness [12, 20] of CMOS inverters. Figure 2 illustrates this relationship. This relationship between energy consumption and probability of correct switching were then consolidated into two laws, the former relates the energy consumption to the probability of correct switching and the latter relates the energy consumption to the noise magnitude.

Law 1: Energy-probability Law:(from [1]) For any fixed technology generation (which determines the capacitance C of a switch) and constant noise magnitude σ, the switching energy $E_{C,\sigma}$ consumed by a probabilistic switch grows with the probability of correctness p. Furthermore, the order of growth of $E_{C,\sigma}$ in p is asymptotically bounded below by an exponential in p.

Law 2: Energy-noise Law:(from [1]) For any fixed probability p of correctness and a fixed technology generation (which determines the capacitance C of the switch), the switching energy $\tilde{E}_{C,p}$ grows quadratically with σ.

These relationships, initially developed in the context of inverters, have been extended to gates such as exclusive-OR and NAND. Further details about the analytical models, simulation and measurement results may be found in [19].

3.2 The Design Principle - Using The E-p Relationship

We recall that in our context, we consider two classes of applications—those which benefit from the probabilistic behavior of computational primitives, and those which can tolerate probabilistic behavior of computational primitives. With the E-p relationship outlined in Section 3.1 and the RaBRAM model in Section 1.1 as background, it is apparent that applications based on probabilistic algorithms may benefit from CMOS devices susceptible to noise [7]. We recall that probabilistic algorithms are implemented on conventional (deterministic) CMOS-based VLSI circuits, by generating random bits from either software or hardware-based pseudo random number generators. To demonstrate this, we have achieved orders of magnitude improvements in energy and performance in the context of probabilistic applications PCMOS technology. Our implementation platform consists of a deterministic host processor—typically a low-energy embedded processor like the StrongArm SA-1100 processor—and a probabilistic application specific co-processor composed of PCMOS devices operated at low voltages. We refer to this platform as a probabilistic system-on-a-chip architecture [8]. We note the energy and performance benefits in the context of applications which can harness probabilistic behavior, arise from two sources (i) the energy efficiency of PCMOS devices when compared to their deterministic counterparts and (ii) the efficiency obtained by implementing probabilistic steps of probabilistic algorithms in inherently probabilistic computational devices. In this context, an important design challenge is to map the probabilistic steps of probabilistic algorithms and the components of algorithms which tolerate perturbations on to inherently probabilistic computational devices. We describe two approaches, rooted in logic and arithmetic to achieve this mapping.

3.2.1 Probabilistic Boolean Logic

In the conventional design automation context, this mapping of application primitives to computational primitives composed of logic gates, is achieved through algorithms rooted in the properties of Boolean logic. Correspondingly, we have studied the probabilistic counterpart of this logic, the probabilistic Boolean logic (PBL) [10]. This logic is composed of Boolean logic primitives with an associated probability of correctness. Well formed probabilistic Boolean formulae (PBF) in this logic—like their deterministic counterparts—can be constructed from the Boolean constants $0, 1$, Boolean variables, and *probabilistic Boolean operators*: *probabilistic disjunction, probabilistic conjunction* and *probabilistic negation*, represented by the symbols $\vee_p$, $\wedge_q$ and $\neg_r$ respectively, where the reals $\frac{1}{2} \leq p, q, r \leq 1$ are the corresponding probability parameters or *probabilities of correctness*. The length of a PBF is defined as the number of operators n in the formula. Given a PBF F, we will use VAR_F to denote the set of variables in F.

If F, G, H denote $(x \vee_p y)$, $(x \wedge_q y)$, and $(\neg_r x)$ respectively, let $T(F_\alpha), T(G_\beta)$ and $T(H_\gamma)$ denote their truth value under the assignments α, β and γ respectively. Then an informal operational approach to assigning or determining "truth" in the case of a PBF is

$$T(F_\alpha) = \begin{cases} T((x \vee y)_\alpha) & \text{with probability } p \\ T(\neg(x \vee y)_\alpha) & \text{with probability } (1-p) \end{cases}$$

$$T(G_\beta) = \begin{cases} T((x \wedge y)_\beta) & \text{with probability } q \\ T(\neg(x \wedge y)_\beta) & \text{with probability } (1-q) \end{cases}$$

$$T(H_\gamma) = \begin{cases} T((\neg x)_\gamma) & \text{with probability } r \\ T((x)_\gamma) & \text{with probability } (1-r) \end{cases}$$

For example, if $p = \frac{3}{4}$ and if F denotes $(x \vee_{\frac{3}{4}} y)$ and α is the assignment $x = 1, y = 0$, then $T(F_\alpha) = T((x \vee y)_\alpha) = T(1 \vee 0) = 1$ with probability $\frac{3}{4}$ with probability $\frac{1}{4}$ it is $T(\neg(x \vee y)_\alpha) = T(\neg(1 \vee 0)) = 0$ and we say that a probabilistic Boolean formula F is satisfied with a probability P under an assignment I, if the value of F is 1 with a probability P under assignment I. For example, if F denotes $(x \vee_{\frac{3}{4}} y)$, then it is satisfied with a probability $P = \frac{3}{4}$ under the assignment $x = 1, y = 0$. Given a PBF the (energy) cost of this formula is the sum of the costs of its constituent operators. As we have seen in Section 3.1, the cost of a probabilistic Boolean operator is related to the probability parameter associated with this operator. Given a probabilistic Boolean formula F, the "underlying" deterministic Boolean formula B is defined to be a formula such that the probabilities of correctness of the individual operators of F is set to 1.

We will now extend this notion of truth with associated probability to arbitrary formulae in PBL. In a *probabilistic Boolean truth table* with $l = 2^k$ $(k > 0)$ rows and three columns, the first column of the n^{th} row contains N, the k bit binary representation of n, $0 \leq n < 2^k$. For example in Figure 3 (ignoring the header rows which are shaded gray) the probabilistic truth table has $2^3 = 8$ rows and 3 columns. For example, the fourth row contains 100, the 3-bit binary representation of 4. The second and the third column of the n^{th} row contain reals $0 \leq p_n, q_n \leq 1$ where $p_n + q_n = 1$. Revisiting the figure, the fourth row contains 3/4 and 1/4 respectively. This represents the fact that $T(F_N)$ is 0 with probability q_n and for the *same formula* for the *same input assignment* it is 1 with probability p_n.

Input	Probabilities	
x y z	Truth Value=1	Truth Value=0
0 0 0	¼	¾
0 0 1	¼	¾
0 1 0	¼	¾
0 1 1	¾	¼
1 0 0	¼	¾
1 0 1	1	0
1 1 0	1	0
1 1 1	1	0

Figure 3: A probabilistic Boolean truth table for the PBF $(((x \wedge_1 y) \vee_1 (x \wedge_1 z)) \vee_1 (y \wedge_{3/4} z))$

If H is a PBF of length 1 or more and F is sub formula of H, say $H = F \vee_p G$, if I is an assignment to variables in H, I' is a *consistent assignment* to variables in F if and only if whenever $x_i \in \text{VAR}_F$, x_i is assigned to the same Boolean constant under the assignments I and I'. Let I be an assignment of the variables of H and let I' and I'' respectively be the consistent assignment of the variables in F, G. Let P_F, P_G and P_H denote the probabilities with which $F_{I'}, G_{I''}$ and H_I are respectively satisfied. Then, axiomatically,

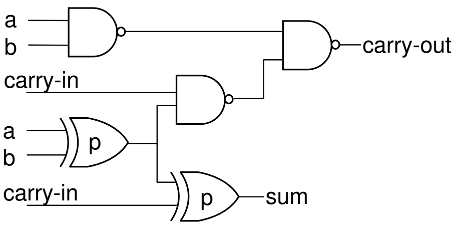
Rule 1: $P_H = (P_F)(P_G)p + (1 - P_F)(P_G)p + (P_F)(1 - P_G)p + (1 - P_F)(1 - P_G)(1 - p)$.

Rule 2: If H denotes $(F \wedge_p G)$, $P_H = (P_F)(P_G)p + (1 - P_F)(P_G)(1-p) + (P_F)(1-P_G)(1-p) + (1-P_F)(1-P_G)(1-p)$.

Rule 3: If H denotes $(\neg_p F)$, $P_H = (P_F)(1-p) + (1-P_F)p$.

We note that the rules mentioned above may be used to determine the truth value of arbitrary probabilistic Boolean formulae for arbitrary assignments. Thus given an arbitrary probabilistic Boolean formula, a probabilistic Boolean truth table may be constructed, which specifies the truth value of this formula under various assignments. Conversely, we have shown that for any probabilistic Boolean truth table, there exists a probabilistic Boolean formula which computes this truth table [10]. Corresponding to the conventional case where any Boolean formula represents a Boolean circuit (which can then be implemented on logic gates constructed from CMOS technology), any PBF represents a *probabilistic* Boolean circuit, which can then be implemented using logic gates constructed from PCMOS technology.

We note that a probabilistic Boolean truth table can specify the probabilistic behavior of the steps of any probabilistic algorithm. It can also be used to quantify the "slack" in the operations of algorithms which can tolerate perturbations. We shall refer to the problem of determining a probabilistic Boolean formula which computes a given probabilistic truth table as *probabilistic logic synthesis* problem. The framework of probabilistic logic synthesis may be utilized to construct probabilistic circuits using PCMOS, thus implementing probabilistic algorithms and algorithms which naturally tolerate perturbations, in an efficient manner.



Input	Output	
	0	1
0 0 0	r	$(1-r)$
0 0 1	$(1-r)$	r
0 1 0	$(1-r)$	r
0 1 1	r	$(1-r)$
1 0 0	$(1-r)$	r
1 0 1	r	$(1-r)$
1 1 0	r	$(1-r)$
1 1 1	$(1-r)$	r

Figure 4: (a) A full adder with probabilistic exclusive-OR gates marked p (b) Probabilistic truth table for the probabilistic Boolean formula $(a \oplus_p (b \oplus_p c))$ or $(a \oplus_r (b \oplus_1 c))$ where $r = 2p^2 - 2p + 1$

3.2.2 Probabilistic Arithmetic and Digital Signal Processing

PBF may be used to construct probabilistic Boolean functions to compute the arithmetic primitives of DSP applications. To do this, we consider the simplest method of addition, the ripple-carry technique of addition. The design of a probabilistic full-adder in a ripple carry adder has been illustrated in Figure 4. The gates of this circuit are probabilistic and using the principles of PBL, the truth table which corresponds to this circuit has been shown in Figure 4.

We recall from Section 3.1 that for a slight sacrifice in the probability of correctness of exclusive-OR gates, significant savings in energy may be obtained. Thus a full adder constructed from probabilistic exclusive-OR gates would be energy efficient when compared to a conventional deterministic and correct full adder. A ripple carry adder composed of such full adders may then be used to implement arithmetic primitives of digital signal processing applications. For example, we have implemented H.264 decoding using such probabilistic adders and one video frame obtained by decoding H.264 video [23] is shown in Figure 5(b). We observe that since the energy investment in each full-adder (which we refer to as the "uniform voltage scaled" case), the probability of errors in bits of lower significance is the same as the probability of bits of a higher significance. But errors in the bits of a higher significance introduce a high magnitude of error in the output. Thus we introduce the concept of non uniform voltage scaling or biased voltage scaling (BIVOS) wherein the circuit components which compute bits of a higher significance are operated at a higher voltage (and hence with a higher energy investment) than the circuit components which compute bits of a lower significance [15]. Thus, bits of a higher significance incur less errors than bits of a lower significance, reducing the expected magnitude of error from an adder. If this principle were to be applied in the design of a finite impulse response (FIR) filter, H.264 decoding may be performed in an energy efficient manner [15] with an acceptable degradation in the quality of the output. This is illustrated in Figure 5(c).

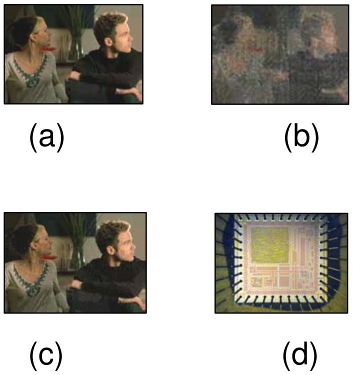


Figure 5: (a) A conventional CMOS implementation and one frame of a video obtained by decoding using the H.264 standard (b) The same image reconstructed in an energy saving mode using uniform voltage scaling yielding a visually distorted image, (c) The same reconstruction using non-uniform voltage scaling using the same energy as the preceding case (d) An example test chip that has been fabricated and tested

4. APPROXIMATE DESIGN

So far, we have considered the relationship between energy and probabilistic computing, defined a new system of mathematical logic which incorporates probability and Boolean logic, and implemented applications using design principles rooted in PBL and probabilistic arithmetic on PCMOS technology, to yield performance and energy benefits. We note that this approach will yield benefits in scaled noise-susceptible future CMOS technology. However the principles of probabilistic design—trading energy for correctness of constituent operations of applications—may be applied in the context of current-day technology generations as well.

4.1 Approximate Arithmetic

In this context, let us define an addition operator $+_\delta$ to be *approximately δ correct*, if for any $0 \le a, b \le n$, let $|(a +_\delta b) - (a + b)| \le \delta$. We wish to distinguish approximately correct addition and probabilistic addition, by noting that the former is deterministic, whereas the latter is probabilistic. That is, for any two fixed inputs, the result of approximate addition (though incorrect) is the same across multiple additions, however in probabilistic addition, across multiple additions, the results may vary for the same inputs. If some property of CMOS devices exists such that an implementation with a higher δ incurs less energy when compared to a lower δ, these arithmetic operations could be used to implement DSP applications and would provide a novel way to trade quality of solution for cost. We have demonstrated that *voltage overscaling*—operating arithmetic circuits at a lower voltage such that their operating frequency violates their critical path delay—is a VLSI technique that can be used to implement approximately correct arithmetic operators [9].

4.2 Implementing Approximate Arithmetic

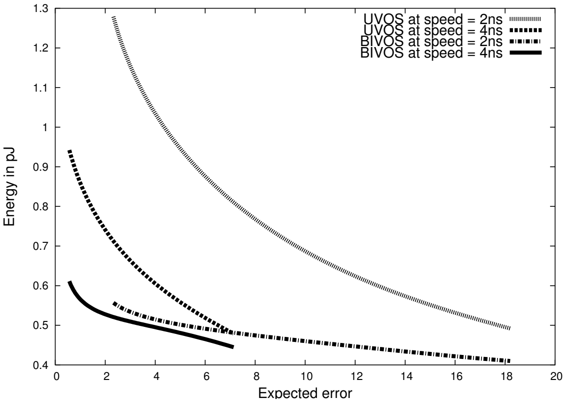
To illustrate the implementation of approximate arithmetic in VLSI, let us consider the case of an 8-bit ripple carry adder, composed of eight full adders. Let these full adders be labeled $\mathrm{FA}_7, \mathrm{FA}_6, \ldots, \mathrm{FA}_0$ with the adder FA_i computing the i^{th} bit of the output. We consider the case where each of these full adders are operated at the same supply voltage, thus incurring a delay d to compute the carry. In the conventionally correct operating scenario, the ripple carry adder would be operated at a frequency $f = 1/8d$, since in the worst case, the critical path delay is $8d$. In the voltage overscaled case, to obtain energy efficiency, each full adder may be operated at a lower supply voltage, such that each of the full adders incur a delay $d' > d$—thus in the worst case, the critical path delay is $8d'$. Now to retain the performance as before, the ripple carry adder is operated at frequency $f = 1/8d > 1/8d'$. In such a scenario, for certain input combinations—for example, the case where 11111111 is added to 00000001 and the carry produced at the least significant position needs to be propagated to the most significant position—the result of the adder would be read before the computation is completed, thereby producing an incorrect result. This is a technique of implementing approximate arithmetic in VLSI and the case where all of the full adders are operated at identical voltage levels is referred to as the uniform voltage scaling case or the UVOS case.

As before, we observe that if bits of a higher significance were erroneous, the magnitude of error introduced in the result would be higher. Thus a non-uniform voltage scaling scheme—the full adders which compute bits of a higher significance are operated at a higher voltage level than the full adders which compute bits of a lower significance—may be envisaged. We refer to this scheme as the *biased voltage scaling* or BIVOS scheme. As a modification of the BIVOS scheme, we may consider a *binned* non-uniform voltage scaling scheme, where the eight full adders may be grouped into a total of $k < 8$ groups or bins—for example, let us consider four groups with each adder with an even index grouped with the adder with the next higher index—and each distinct group is operated with a distinct supply voltage.

An approximate ripple carry adder ARCA consists of k full-adders $\mathrm{FA}_{k-1}, \mathrm{FA}_{k-2}, \ldots, \mathrm{FA}_0$. The inputs to FA_i are the bits at position i in the two numbers. Whenever $i > 0$, the carry bit from FA_{i-1} is an input to FA_i. The full adders are grouped into l bins where the adders in bin j are all operated at the same voltage V_j. While using BIVOS approach, voltage V_j associated with bin B_j is greater than the voltage V_i associated with bin B_i is operating, whenever $j > i$ and $0 \leq i, j \leq k-1$. Let d_i be the total *delay* associated with bin B_i. Then the *critical path* of a circuit realization of an ARCA is $\rho = d_0 + d_1 + \cdots + d_{l-1}$.

Let X be the *correct* sum of the inputs to an ARCA where the correct sum is computed after ρ cycles have elapsed. Similarly, let $\hat{X}$ be the value eagerly consumed from the sum of the ARCA after $\epsilon < \rho$ cycles. Then, given a set of sums $\hat{X_1}, \hat{X_2}, \ldots, \hat{X_n}$, we define with associated correct values $X_1, X_2 \ldots, X_n$ respectively, the *expected error percentage* $\mathbf{E} = \frac{100}{n} \sum_{i=1}^{n} |\hat{X_i} - X_i|$. In the rest of the paper we will use the term *expected error* for *expected error percentage* for convenience.

To experimentally determine the energy consumption and the expected error magnitude of an ARCA, we simulate the behavior of the adder. The simulation framework is based on individual gate-level HSPICE simulations, feeding a C-based simulation of an adder. We strengthen this approach by considering detailed post-layout models designed using Cadence Tools Virtuoso Layout editor with NCSU_TechLib_TSMC02 (180 nm) which include parasitics and the overheads involved in creating multiple voltage levels in the ripple carry adder (RCA) circuitry. The range of voltages in which the full adders are operated is 0.7V to 1.8V. While static power is included in individual HSPICE simulations, the C-based simulator only adds up power consumed by gates that switch thus ignoring static power consumed by gates that do not switch. However, the resulting inaccuracy is small, as leakage is small in 180 nm technology. Furthermore, the C-based software simulator has been validated to be within a margin of 2% by two complete HSPICE simulations of the approximate ripple carry adder, one with a set of high supply voltages over the various bins and one with a set of low supply voltages. One of the metrics that will be used is the energy-delay product denoted as EDP.



[ht]

Figure 6: The E-E relationship for a 16-bit ARCA using the BIVOS and the UVOS schemes, where ρ = 2 ns and 4 ns

4.2.1 The Relationship between Energy, Error and Delay in an ARCA

The relationship between energy consumed (E) to compute a 16-bit addition and its associated expected error $\mathbf{E}$, henceforth referred to as the E-$\mathbf{E}$ relationship, is shown in Figure 6 for the BIVOS as well as the UVOS cases. At an expected error of 17%, a BIVOS adder operating in 2 ns is 3.25X faster and consumes 3.8X lower energy when compared to a conventional adder, i.e., the BIVOS based adder is 12.3X more efficient in terms of EDP. In Figure 7, we present the relationship between energy (E) and delay (ϵ), henceforth referred to as the E-ϵ relationship. In this figure, we present the relationship for two different error values, 5% and 15%.

4.3 Comparing Alternate Adder Architectures and the E-E Relationship

We will now compare the behavior of an ARCA to alternate adder architectures, specifically approximate carry skip adder (ACSA) and approximate carry look-ahead adder (ACLA). Consider the manner in which the behavior of adders in Figure 8 vary as we increase the operating speed from 1 ns to 4 ns. The basic observation is that for a fixed

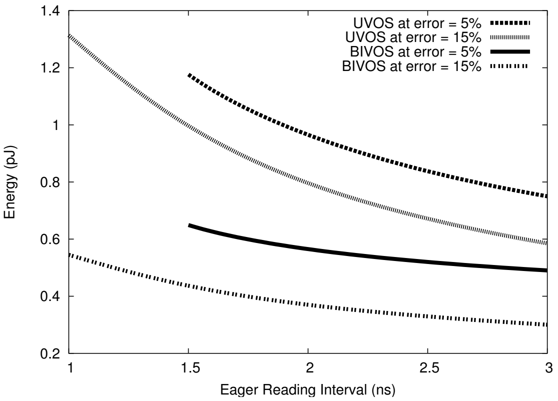


Figure 7: The relationship between energy and delay as the admitted error varies in an ARCA

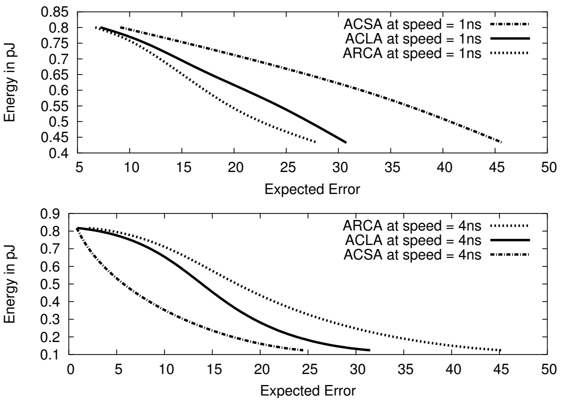


Figure 8: The E-E relationship for the ARCA, ACLA and ACSA architectures for an eager reading interval of 1 ns and 4 ns

amount of error, while an ARCA is indeed the most energy efficient adder design at 1 ns, its relative advantage decreases with increasing ϵ. This is because when ϵ increases, then the ACSA and ACLA propagate the carry information more than an ARCA for a relatively small additional energy investment. And thus the ACSA and ACLA have increasingly lower error. *Thus, at higher ϵ when the advantage of the additional circuitry is utilized completely, the* ACSA *performs the best, followed by the* ACLA*, with the* ARCA *trailing both of them.* This represents a reversal of relative energy and EDP efficiency of the three architectures in going from an ϵ equals 1 ns to ϵ equals 4 ns.

4.3.1 Trading Perceived Value for Energy Savings Through an ARCA

One of our key goals is to demonstrate the value of approximate adder architectures in the domain of video (image signal) processing. We do this by using the DFT on images and then reconstruct the original image using its inverse. The significance of approximate arithmetic as a viable approach to realizing energy savings is apparent from Figure 9. In this figure, the images labeled as BIVOS 1, BIVOS 2, UVOS 1 and UVOS 2 correspond to different voltage biasing schemes used for their respective computation. In Figure 10 going from a completely correct computation (which is labeled Conventional to that using BIVOS (labeled BIVOS 1 and BIVOS 2 in Figure 10) an EDP savings of 1.3X and 1.75X respectively was achieved with an associated *signal to noise ratio* (SNR) of 29.41 dB and 1.4 dB, while preserving the computational speed or performance. In contrast and with similar EDP values, a UVOS based approach (labeled as UVOS 1 and UVOS 2 in Figure 10) results in obviously unacceptable images with an associated SNR of −2.7 dB and −3.8 dB respectively.

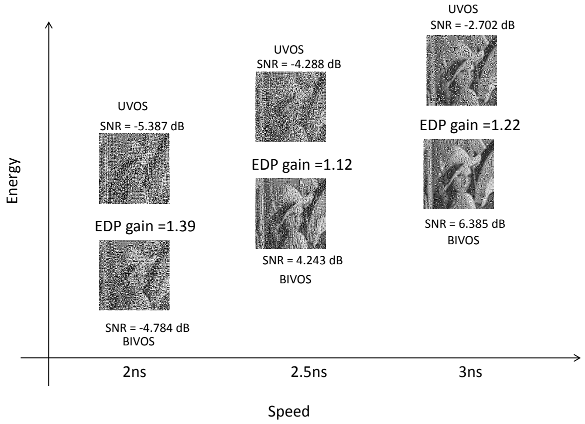


Figure 9: The EDP advantages of BIVOS over UVOS in processing an image where the EDP gain of BIVOS based approach over a UVOS based approach is shown in the figure between the pairs of reconstructed images

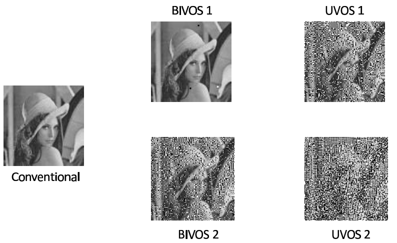


Figure 10: The image obtained after applying DFT and then applying inverse-DFT using conventional correctly operating adders, adders with BIVOS and adders with UVOS

5. FUTURE DIRECTIONS

We have surveyed our work on probabilistic and approximate design, whose central thrust is the design of systems with components susceptible to perturbations and the compromise in the quality of solution for cost savings. Broadly future research may be pursued in the domain of applications, technology attributes and design principles.

5.1 Applications

In the domain of applications, using probabilistic and approximate design to implement a larger class of more complex applications, such as the efficient implementation of audio and speech processing may be explored. Speech processing and enhancement, if performed with ultra low-energy consumption, would be extremely valuable in the context of bio-prostheses such as completely in canal hearing aids. In addition, the design and implementation of graphics processing using probabilistic design, can enable the implementation of low-energy multimedia devices and educational tablets. Such mobile educational tablets, together with innovative pedagogical content have a great potential to supplement classroom teaching in resource-constrained areas of the world. Low power image processors can also be used in mobile medical diagnostic devices, enabling their wide deployment for rapid screening for various diseases which can then be treated in an early and effective manner.

5.2 Technology and Architecture

In the context of probabilistic arithmetic, we have considered *temporal* perturbations and demonstrated the use of probabilistic design towards using such devices with perturbations to implement signal processing and other applications. A dual of this approach is to consider devices with *spatial* perturbations and adapt our techniques accordingly. This will be useful towards mitigating the effects of parameter variations within a die. In addition, analogous to the energy-delay relationship and the E-p relationship in CMOS, other quality-cost relationships can be studied in novel non-CMOS materials such as molecular electronics. In the context of approximate designs, we have considered the problem of allocating voltage levels to circuit components, to adjust the speed of operation of various circuit components and achieve a good trade-off between energy consumption and the quality of solution. In the context of a field programmable gate array, whose configurable blocks have various speeds of operation, a dual approach would be the assignment of circuit components to configurable blocks, to maximize quality of solution. Thus approximate design can be used in conjunction with techniques such as adaptive body bias [34] to address the effects of parameter variations in circuits.

5.3 Design Principles and Tools

In the context of approximate arithmetic circuits, we have considered the problem of the distribution of the energy budget among the full adders in a ripple carry adder, to minimize error. We have also empirically studied various schemes of distributing the energy budget among the subcomponents in alternate adder architectures. This approach can be generalized to arbitrary circuits. In particular, arithmetic circuits which implement digital signal processing primitives, such as the fast Fourier transform (FFT), are candidates of interest. This problem can be informally stated as follows: Given a circuit, which can be modeled as a directed acyclic graph—the vertices of the graph represent circuit components such as adders—determine the optimal allocation of energy among the vertices such that the overall error magnitude is minimized. This global optimization of energy allocation in arithmetic circuits such as filters would yield a better quality of solution when compared to the case where locally optimized circuit components are composed. The theory of PBL can be further developed to derive results of independent mathematical interest. Furthermore, *probabilistic logic synthesis* for automatic synthesis and optimization of probabilistic circuits can be explored. Design automation tools to explore the design space of probabilistic and approximate arithmetic circuits will be useful towards implementing such circuits in an efficient way.

6. ACKNOWLEDGMENTS

This work was supported in part by the US Defense Advanced Research Projects Agency (DARPA) under seedling contract number F30602-02-2-0124, the DARPA ACIP program under contract FA8650-04-C-7126 through a subcontract from USC-ISI, and an award from the Intel Corporation. Krishna Palem wishes to thank the Moore distinguished faculty fellow program at the California Institute of Technology, and the Canon distinguished professorship program of the Nanyang Technological University at Singapore, during the 2007 academic year, which enabled pursuing this work in part. We acknowledge the contributions of our colleagues Dr. Bilge Akgul, Suresh Cheemalavagu, Jason George, Dr. Pinar Korkmaz, Harry Marr and Dr. Balasubramanian Seshasayee, who are co-authors and collaborators in our work which we have surveyed. We wish to acknowledge the support and suggestions of Professor James Meindl, whose paper [24] inspired the structure of this article's title. The concept of Figure 1 is based on Professor Richard Karp's Turing award lecture [17].

7. REFERENCES

[1] B. Akgul, L. Chakrapani, P. Korkmaz, and K. Palem. Probabilistic CMOS technology: A survey and future directions. In *Proceedings of the IFIP International Conference on Very Large Scale Integration*, pages 1–6, Oct. 2006.

[2] R. I. Bahar, J. Mundy, and J. Chen. A probabilistic-based design methodology for nanoscale computation. In *Proceedings of the 2003 IEEE/ACM International Conference on Computer-aided Design*, pages 480–486, 2003.

[3] S. Borkar. Exponential challenges, exponential rewards - The future of Moore's law. In *Proceedings of the IFIP International Conference on Very Large Scale Integration (VLSI-SoC)*, page 2, 2003.

[4] S. Borkar. Designing reliable systems from unreliable components: The challenges of transistor variability and degradation. *IEEE Micro*, 25(6):10–16, 2005.

[5] 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 Annual Conference on Design Automation (DAC)*, pages 338–342, 2003.

[6] K. Bowman, J. Tschanz, N. S. Kim, J. Lee, C. Wilkerson, S.-L. Lu, T. Karnik, and V. De. Energy-efficient and metastability-immune timing-error detection and recovery circuits for dynamic variation tolerance. *IEEE International Conference on Integrated Circuit Design and Technology and Tutorial*, pages 155–158, 2008.

[7] L. N. B. 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*, pages 1110–1115, Mar. 2006.

[8] L. N. B. Chakrapani, P. Korkmaz, B. E. S. Akgul, and K. V. Palem. Probabilistic system-on-a-chip architectures. *ACM Transactions on Design Automation of Electronic Systems*, 12(3):1–28, 2007.

[9] L. N. B. Chakrapani, K. K. Muntimadugu,

L. Avinash, J. George, and K. V. Palem. Highly energy and performance efficient embedded computing through approximately correct arithmetic: A mathematical foundation and preliminary experimental validation. In *Proceedings of the IEEE/ACM International Conference on Compilers, Architecture, and Synthesis of Embedded Systems*, 2008.

[10] L. N. B. Chakrapani and K. V. Palem. A probabilistic Boolean logic and its meaning. *Rice University, Department of Computer Science Technical Report*, (TR-08-05), June 2008.

[11] 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 International Conference on Solid State Devices and Materials*, pages 402–403, Sept. 2004.

[12] 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 (VLSI-SoC)*, pages 452–457, 2005.

[13] V. De and S. Borkar. Low power and high performance design challenges in future technologies. In *Proceedings of the 10th Great Lakes symposium on VLSI*, pages 1–6, New York, NY, USA, 2000. ACM.

[14] D. Ernst, N. S. Kim, S. Das, S. Pant, T. Pham, R. Rao, C. Ziesler, D. Blaauw, T. Austin, and T. Mudge. Razor: A low-power pipeline based on circuit-level timing speculation. In *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 7–18, Oct. 2003.

[15] J. George, B. Marr, B. E. S. Akgul, and K. Palem. Probabilistic arithmetic and energy efficient embedded signal processing. In *Proceedings of the The IEEE/ACM International Conference on Compilers, Architecture, and Synthesis for Embedded Systems*, pages 158–168, 2006.

[16] ITRS. International technology roadmap for semiconductors, 2007.

[17] R. M. Karp. Combinatorics, complexity, and randomness. *Communications of the ACM*, 29(2):98–109, 1986.

[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. *Probabilistic CMOS (PCMOS) in the Nanoelectronics Regime*. PhD thesis, Georgia Institute of Technology, 2007.

[20] 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*, 45(4B):3307–3316, Apr. 2006.

[21] H. Leff and A. Rex, editors. *Maxwell's Demon: Entropy, Information, Computing.* Princeton University Press, 1990.

[22] H. Li, J. Mundy, W. Paterson, D. Kazazis, A. Zaslavsky, and R. Bahar. Thermally-induced soft errors in nanoscale CMOS circuits. *IEEE International Symposium on Nanoscale Architectures*, pages 62–69, 2007.

[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. Low power microelectronics: Retrospect and prospect. *Proceedings of The IEEE*, 83:619–635, Apr. 1995.

[25] K. V. Palem. Energy aware algorithm design via probabilistic computing: From algorithms and models to Moore's law and novel (semiconductor) devices. In *Proceedings of the IEEE/ACM International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, pages 113–117, 2003.

[26] K. V. Palem. Proof as experiment: Probabilistic algorithms from a thermodynamic perspective. In *Proceedings of the International Symposium on Verification (Theory and Practice),*, 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] K. V. Palem, S. Cheemalavagu, P. Korkmaz, and B. E. Akgul. Probabilistic and introverted switching to conserve energy in a digital system. *US Patent*, (20050240787), 2005.

[30] K. V. Palem, P. Korkmaz, and K.-S. Y. Z.-H. Kong. Probabilistic CMOS (PCMOS) logic for nanoscale circuit design. In *International Solid State Circuits Conference: Advanced Solid-State Circuits Forum*, 2009.

[31] M. O. Rabin. Probabilistic algorithms. In J. F. Traub, editor, *Algorithms and Complexity, New Directions and Recent Trends*, pages 29–39. Academic Press, 1976.

[32] R. Solovay and V. Strassen. A fast monte-carlo test for primality. *SIAM Journal on Computing*, pages 84–85, 1977.

[33] J. M. Tour and D. K. James. Molecular electronic computing architectures: A review. In I. Goddard, W. A., D. W. Brenner, S. E. Lyshevski, and G. J. Iafrate, editors, *Handbook of Nanoscience, Engineering and Technology, Second Edition*, pages 5.1–5.28. CRC Press, New York, 2007.

[34] J. W. Tschanz, J. T. Kao, S. G. Narendra, R. Nair, D. A. Antoniadis, A. P. Chandrakasan, and V. De. Adaptive body bias for reducing impacts of die-to-die and within-die parameter variations on microprocessor frequency and leakage. *IEEE Journal Of Solid-State Circuits*, pages 1396–1402, 2002.